AN ESTIMATOR FOR AN ANTI-AIRCRAFT GUN FIRE CONTROL SYSTEM

John Michael Parr

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93940

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AN ESTIMATOR FOR AN ANTI-AIRCRAFT GUN FIRE CONTROL SYSTEM

by

John Michael Parr

December 1974

Thesis Advisor: D. E. Kirk

Approved for public release; distribution unlimited.

T165278

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
AN ESTIMATOR FOR AN ANTI-AIRCRAFT
GUN FIRE CONTROL SYSTEM

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis;
December 1974

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
John Michael Parr

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE
December 1974

13. NUMBER OF PAGES
151

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Naval Postgraduate School
Monterey, California 93940

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Kalman Filter; Tracking; Rotated Coordinate System
Monte-Carlo Simulation; Adaptive Filtering

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Kalman filtering techniques utilizing a rotated coordinate system were applied to tracking problems encountered in fire control systems. Two models of target motion were considered: a constant-velocity model and a model which assumes correlated random accelerations. Estimators derived from these models were evaluated using Monte-Carlo simulations of constant-velocity and maneuvering targets. An algorithm

Block 20 - ABSTRACT (Cont.)

developed to calculate prediction accuracy data for time intervals based on an approximation of the time of flight for a 5"/54 Caliber projectile was used to obtain prediction accuracy statistics for evaluating estimator performance. A track generating program was developed to produce maneuvering target state trajectories in either cartesian or spherical coordinates for use in the Monte-Carlo simulations. Adaptive filters were developed to take advantage of the best features of the two basic estimators and Monte-Carlo simulation of the adaptive filters was performed.

An Estimator for An Anti-Aircraft
Gun Fire Control System

by

John Michael Parr
Lieutenant, United States Navy
B.S.E.E., Auburn University, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1974

Thesis
[illegible]
c.1

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93940

ABSTRACT

Kalman filtering techniques utilizing a rotated coordinate system were applied to tracking problems encountered in fire control systems. Two models of target motion were considered: a constant-velocity model and a model which assumes correlated random accelerations. Estimators derived from these models were evaluated using Monte-Carlo simulations of constant-velocity and maneuvering targets. An algorithm developed to calculate prediction accuracy data for time intervals based on an approximation of the time of flight for a 5"/54 Caliber projectile was used to obtain prediction accuracy statistics for evaluating estimator performance. A track generating program was developed to produce maneuvering target state trajectories in either cartesian or spherical coordinates for use in the Monte-Carlo simulations. Adaptive filters were developed to take advantage of the best features of the two basic estimators and Monte-Carlo simulation of the adaptive filters was performed.

TABLE OF CONTENTS

I. INTRODUCTION -------------------------------------- 10

A. ALTERNATIVE ESTIMATION TECHNIQUES ----------- 10

B. BACKGROUND MATERIAL ------------------------- 12

II. THE ROTATED CARTESIAN COORDINATE SYSTEM --------- 15

A. MOTIVATION FOR ROTATING THE COORDINATE SYSTEM -------------------------------------- 15

1. Correlation of Measurement Noise -------- 15

2. Modeling of Random Forcing Inputs ------- 18

B. COORDINATE SYSTEM ROTATION ----------------- 21

III. FILTER DEVELOPMENT ------------------------------ 23

A. THE CONSTANT-VELOCITY ($1/S^2$) MODEL ---------- 23

B. THE CORRELATED RANDOM ACCELERATION MODEL ---- 28

IV. MONTE-CARLO SIMULATION --------------------------- 32

A. TEST TRAJECTORIES --------------------------- 32

1. The Short-Range, Constant-Velocity Trajectory ------------------------------ 32

2. The Long-Range, Constant-Velocity Trajectory ------------------------------ 32

3. The Short-Range Trajectory With Maneuver- 33

4. The Long-Range Trajectory With Maneuver - 33

5. Other Trajectories ---------------------- 33

B. ACCELERATIONS ASSUMED ---------------------- 34

C. SIMULATION RESULTS -------------------------- 35

D. COMPARISON OF PREDICTION ACCURACY AND ESTIMATION ERROR STATISTICS ----------------- 41

V. ADAPTIVE FILTERS ---------------------------------- 44
A. THE SWITCH-ON-RANGE ADAPTIVE FILTER --------- 44
B. SWITCH-ON-RESIDUALS ADAPTIVE FILTER #1 ------ 45
C. SWITCH-ON-RESIDUALS ADAPTIVE FILTER #2 ------ 48
D. THE COMPROMISE FILTER ---------------------- 49
E. SUMMARY OF ADAPTIVE FILTER DEVELOPMENT ------ 49
VI. ADAPTIVE FILTER SIMULATION ---------------------- 51
A. TEST TRAJECTORIES --------------------------- 51
B. SIMULATION RESULTS -------------------------- 52
VII. SUMMARY AND CONCLUSIONS ------------------------- 54
A. TEST RESULT SUMMARY ------------------------- 54
1. Simulation Result Summary for the Two Basic Models ---------------------------- 54
2. Adaptive Filter Simulation Summary ------ 55
B. SUGGESTIONS FOR FURTHER INVESTIGATION ------- 56
C. CONCLUSIONS --------------------------------- 57
APPENDIX A THE MONTE-CARLO SIMULATION PROGRAM -------- 58
A.1 Program Description ----------------------- 58
A.2 The Monte-Carlo Simulation Program Listing - 59
A.3 Flow Chart of the Monte-Carlo Simulation Program ---------------------------------- 89
A.4 Alphabetical Listing of Key Variables Used in the Monte-Carlo Program ----------------- 105
A.5 A List of IBM System/360 Source Library Subroutines Used by the Monte-Carlo Program- 106
APPENDIX B PREDICTION ACCURACY CALCULATION ----------- 107
APPENDIX C A TARGET TRAJECTORY GENERATING PROGRAM ---- 111
C.1 Description of the Program ---------------- 111

C.2 Relationship Between Velocity, Period of Maneuver, Amplitude of Maneuver, and Acceleration -------------------------------- 119

C.3 Examples of Track Generating Program Use --- 120

C.4 The Track Generating Program Listing ------- 126

APPENDIX D ADAPTIVE FILTER SUBROUTINES --------------- 134

BIBLIOGRAPHY --- 150

INITIAL DISTRIBUTION LIST ------------------------------ 151

LIST OF TABLES

Table

I COMPARISON OF MAXIMUM ACCELERATIONS ------------ 20

II CONSTANT-VELOCITY MODEL SIMULATION RESULTS ------ 36

III CORRELATED-RANDOM-ACCELERATION MODEL WITH α=0.3 - 37

IV CORRELATED-RANDOM-ACCELERATION MODEL WITH α=0.5 - 38

V FILTER PERFORMANCE RANKINGS --------------------- 40

VI SUMMARY OF ADAPTIVE FILTERS --------------------- 50

VII MONTE-CARLO SIMULATION RESULTS FOR ADAPTIVE FILTERS -- 53

VIII TRACK GENERATING PROGRAM OUTPUT FOR EXAMPLE 1 --- 123

IX TRACK GENERATING PROGRAM OUTPUT FOR EXAMPLE 2 --- 125

LIST OF FIGURES

Figure

1. THE ROTATED CARTESIAN COORDINATE SYSTEM --------- 11

2. THE ROTATED XY PLANE MANEUVER SHOWING ACCELERATION COMPONENTS AT POINT P (BEFORE ROTATION) AND POINT P' (AFTER ROTATION) --------- 19

3. ROTATED COORDINATE SYSTEM GEOMETRY IN RELATION TO FIXED CARTESIAN COORDINATES ------------------ 22

4. PLOT OF TIME OF FLIGHT VS. RANGE FOR A 5"/54 CAL. PROJECTILE ---------------------------------- 108

5. THE PLANAR MANEUVER ----------------------------- 112

6. ROTATION OF THE TRACK IN ELEVATION (Vertical Plane) -- 115

7. ROTATION OF THE TRACK AXIS IN BEARING ----------- 116

8. VELOCITY VECTOR BEFORE (A) AND AFTER (B) ROTATION THROUGH ANGLE B ------------------------ 117

I. INTRODUCTION

A. ALTERNATIVE ESTIMATION TECHNIQUES

Recently the Gun-Fire Control System MK 86 has been the subject of several studies done at the Naval Postgraduate School involving the use of Kalman filtering techniques in a gun-fire-control system [1],[2],[4]. This thesis examines some possible alternatives to the techniques used in GFCS MK 86 and makes comparisons to the GFCS MK 86 filter.

One proposed alternative to GFCS MK 86 procedure is the use of a rotated cartesian coordinate system. The coordinate system is rotated about the vertical (z) axis so that the x axis is aligned with the initial measured target bearing. Figure 1 illustrates the proposed coordinate system and its relationship to the fixed cartesian and spherical coordinate systems when noiseless measurements are assumed. In the general case, when the measurements are noisy, the x-axis will not be aligned exactly with the true target bearing as indicated in Fig. 1 but the error is expected to be small. The advantages gained by rotating the coordinate system are diminished correlation of the measurement noise and more accurate modeling of random forcing inputs. These attributes are discussed quantitatively in Chapter II.

In Chapter III estimators based on two models of target motion are developed: a constant-velocity model similar to one used in GFCS MK 86 [1] but utilizing the rotated coordinate system, and a constant-velocity model with a coloring

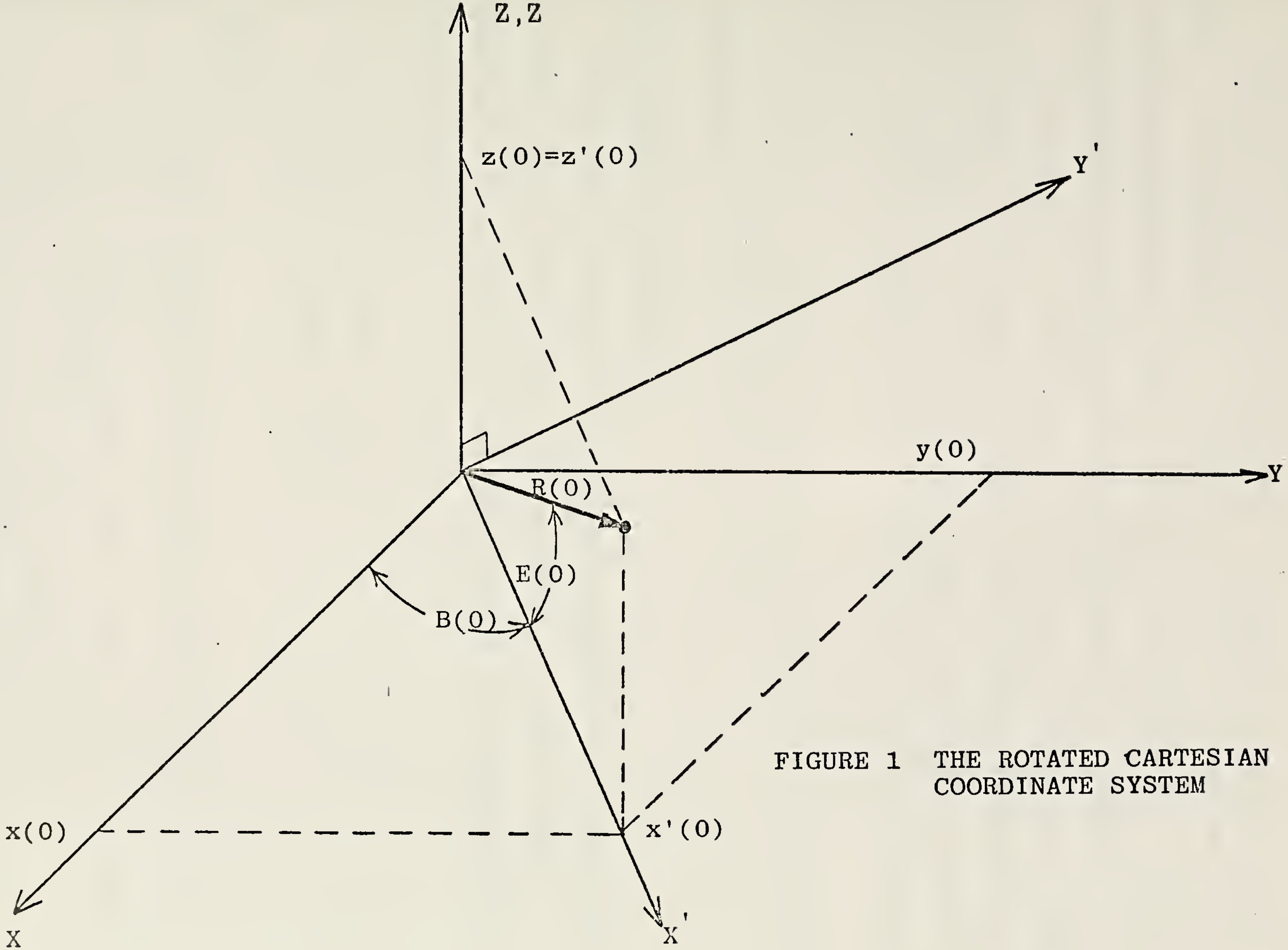


FIGURE 1 THE ROTATED CARTESIAN COORDINATE SYSTEM

pre-filter similar to that described by McKinley [4] also using the rotated coordinate system.

Monte-Carlo simulations were performed in order to evaluate the performance of several variations of the two basic filters developed in Chapter III. Descriptions and results of the Monte-Carlo simulations are presented in Chapter IV.

In Chapter V the results of the Monte-Carlo tests reported in Chapter IV are used to develop two methods of making the filter adaptive to changing situations. Monte-Carlo simulation results for the adaptive filters are reported in Chapter VI.

The final chapter summarizes the results of the various simulations and presents conclusions.

B. BACKGROUND MATERIAL

The model used for the system, composed of a target and a radar which makes noisy measurements of target range, bearing and elevation is approximated by the linear difference equations

$$\underset{\sim}{x}(k+1) = \underset{\sim}{\Phi}\underset{\sim}{x}(k) = \underset{\sim}{\Gamma}\underset{\sim}{w}(k) \qquad (1.1)$$

$$\underset{\sim}{z}(k) = \underset{\sim}{H}\underset{\sim}{x}(k) + \underset{\sim}{v}(k) \qquad (1.2)$$

where

$\underset{\sim}{x}(k)$ is the n-dimensional state vector at time t=kT

$\underset{\sim}{w}(k)$ is the p-dimensional random forcing vector at time t=kT

$\underset{\sim}{\Phi}(k)$ is the n×n state transition matrix

$\underset{\sim}{\Gamma}(k)$ is the n×p matrix which related the random forcing terms to the state vector

$\underset{\sim}{z}(k)$	is the m-dimensional vector of measurements taken at time t=kT
$\underset{\sim}{H}$	is the m×n observation matrix
$\underset{\sim}{v}(k)$	is the m-dimensional vector of measurement noise
T	is the sampling period
k	is a non-negative integer which defines the current sample.

The assumed random process statistics are summarized by the following equations.

$$E\{\underset{\sim}{v}(k)\} = E\{\underset{\sim}{w}(k)\} = \underset{\sim}{0} \text{ for all } k \tag{1.3}$$

$$E\{\underset{\sim}{v}(k)\underset{\sim}{v}^T(j)\} = \underset{\sim}{R}(k)\delta_{jk} \tag{1.4}$$

$$\underset{\sim}{\Gamma}(k)E\{\underset{\sim}{w}(k)\underset{\sim}{w}^T(j)\}\Gamma^T(k) = \underset{\sim}{Q}(k)\delta_{jk} \tag{1.5}$$

$$E\{\underset{\sim}{v}(k)\underset{\sim}{w}(j)\} = \underset{\sim}{0} \text{ for all } j,k. \tag{1.6}$$

where

E	is the expectation operator
δ_{jk}	is the Kronecker delta function
$\underset{\sim}{R}(k)$	is the m×m covariance of measurement noise matrix
$\underset{\sim}{Q}(k)$	is the n×n state excitation covariance matrix.

The discrete Kalman filter equations are [5]

$$G(k) = \underset{\sim}{P}(k/k-1)\underset{\sim}{H}^T[\underset{\sim}{H}\underset{\sim}{P}(k/k-1)\underset{\sim}{H}^T + \underset{\sim}{R}(k)]^{-1} \tag{1.7}$$

$$\underset{\sim}{P}(k/k) = [\underset{\sim}{I} - \underset{\sim}{G}(k)\underset{\sim}{H}]\underset{\sim}{P}(k/k-1) \tag{1.8}$$

$$\underset{\sim}{P}(k/k-1) = \underset{\sim}{\Phi}\underset{\sim}{P}(k-1/k-1)\underset{\sim}{\Phi}^T + \underset{\sim}{Q}(k-1) \tag{1.9}$$

$$\underset{\sim}{\hat{X}}(k/k-1) = \underset{\sim}{\Phi}\underset{\sim}{\hat{X}}(k-1/k-1) \tag{1.10}$$

$$\underset{\sim}{\hat{X}}(k/k) = \underset{\sim}{\hat{X}}(k/k-1) + \underset{\sim}{G}(k)[\underset{\sim}{Z}(k) - \underset{\sim}{H}\underset{\sim}{\hat{X}}(k/k-1)] \tag{1.11}$$

where the notation (k/k-1) denotes the calculation of a quantity at time t=kT based on information up to and

including time $t=(k-1)T$. The previously undefined matrices which appear in the Kalman filter equations are defined as follows:

$\underset{\sim}{G}(k)$ - the n×m Kalman filter gain matrix,

$\underset{\sim}{I}$ - the n×n identity matrix,

$\underset{\sim}{P}(k/k)$ - the n×n covariance of estimation error matrix,

$\underset{\sim}{P}(k/k-1)$ - the one-sample prediction error covariance matrix,

$\underset{\sim}{\hat{X}}(k/k)$ - the n×1 state estimate vector,

$\underset{\sim}{\hat{X}}(k/k-1)$ - the n×1 state prediction vector.

II. THE ROTATED CARTESIAN COORDINATE SYSTEM

A. MOTIVATION FOR ROTATING THE COORDINATE SYSTEM

1. Correlation of Measurement Noise

When radar measurements taken in spherical coordinates are converted to cartesian coordinates for filter processing the measurement noise becomes correlated in the process.

The measurements in spherical coordinates are given by

$$Z_1(k) = r(k) + n_1(k) \tag{2.1}$$

$$Z_2(k) = B(k) + n_2(k) \tag{2.2}$$

$$Z_3(k) = E(k) + n_3(k) \tag{2.3}$$

where $r(k)$, $B(k)$ and $E(k)$ are true target range, bearing and elevation respectively and $n_1(k)$, $n_2(k)$ and $n_3(k)$ are the corresponding measurement noise components. It is assumed that $n_1(k)$, $n_2(k)$ and $n_3(k)$ are mutually uncorrelated and that

$$E\{n_1(k)\} = E\{n_2(k)\} = E\{n_3(k)\} = 0$$

$$E\{n_1^2(k)\} = \sigma_r^2, \ E\{n_2^2(k)\} = \sigma_B^2, \ E\{n_3^2(k)\} = \sigma_E^2 \ ,$$

$$k = 0,1,2,3,.....$$

If the coordinate converted measurements are written as

$$Z_1'(k) = x(k) + v_1(k)$$

$$Z_2'(k) = y(k) + v_2(k)$$

$$Z_3'(k) = z(k) + v_3(k)$$

it can be shown that [2]

$$E\{\underset{\sim}{v}(k)\} = \underset{\sim}{0}$$

$$E\{\underset{\sim}{v}(k)\underset{\sim}{v}^T(k)\} = \underset{\sim}{R}(k) \tag{2.4}$$

where the entries of the covariance matrix of measurement error $\underset{\sim}{R}(k)$ are

$$\begin{aligned} R_{11}(k) = \; & r^2(k)\sigma_E^2\cos^2B(k)\sin^2E(k) \\ & + r^2(k)\sigma_B^2\sin^2B(k)\cos^2E(k) \\ & + r^2(k)\sigma_B^2\sigma_E^2\sin^2B(k)\sin^2E(k) \\ & + \sigma_r^2\cos^2B(k) \end{aligned} \tag{2.5}$$

$$\begin{aligned} R_{12}(k) = \; & r^2(k)\cos B(k)\sin B(k)\sin^2E(k)[\sigma_E^2-\sigma_B^2\sigma_E^2] \\ & + \sin B(k)\cos B(k)\cos^2E(k)[\sigma_r^2-r^2(k)\sigma_B^2] \end{aligned} \tag{2.6}$$

$$R_{13}(k) = \cos B(k)\sin E(k)\cos E(k)[\sigma_r^2-r^2(k)\sigma_E^2] \tag{2.7}$$

$$R_{21}(k) = R_{12}(k) \tag{2.8}$$

$$\begin{aligned} R_{22}(k) = \; & r^2(k)\sigma_E^2\sin^2B(k)\sin^2E(k) \\ & + r^2(k)\sigma_B^2\cos^2B(k)\cos^2E(k) \\ & + r^2(k)\sigma_B^2\sigma_E^2\cos^2B(k)\sin^2E(k) \\ & + \sigma_r^2\sin^2B(k)\cos^2E(k) \end{aligned} \tag{2.9}$$

$$R_{23}(k) = \sin B(k)\sin E(k)\cos E(k)[\sigma_r^2-r^2(k)\sigma_E^2] \tag{2.10}$$

$$R_{31}(k) = R_{13}(k) \tag{2.11}$$

$$R_{32}(k) = R_{23}(k) \tag{2.12}$$

$$R_{33}(k) = r^2(k)\sigma_E^2\cos^2E(k) + \sigma_r^2\sin^2E(k) \tag{2.13}$$

The GFCS MK 86 filter assumes zero mean uncorrelated noise [1], however, equations (2.5) through (2.13) indicate that in general the measurement noise is correlated.

The effect of the correlation is diminished in the rotated cartesian coordinate system, since for an inbound target the bearing will be zero or near zero. If $B(k) = 0$ is assumed,

$$\underset{\sim}{R}(k) = \begin{bmatrix} r^2(k)\sigma_E^2 \sin^2 E(k) + \sigma_r^2 & 0 & \sin E(k)\cos E(k)[\sigma_r^2 - r^2(k)\sigma_E^2] \\ 0 & r^2(k)\sigma_B^2[\cos^2 E(k) + \sigma_E^2 \sin^2 E(k)] & 0 \\ \sin E(k)\cos E(k)[\sigma_r^2 - r^2(k)\sigma_E^2] & 0 & r^2(k)\sigma_E^2 \cos^2 E(k) + \sigma_r^2 \sin^2 E(k) \end{bmatrix} \tag{2.14}$$

Although the measurement noise is still correlated, the dependence on target bearing has been eliminated. For the case of the low elevation target, where the measurement noise is expected to be relatively high due to poor tracking conditions in the radar, the approximations $\sin E(k) \simeq 0$ and $\cos E(k) \simeq 1$ are valid and

$$\underset{\sim}{R}(k) \simeq \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & r^2(k)\sigma_B^2 & 0 \\ 0 & 0 & r^2(k)\sigma_E^2 \end{bmatrix} \tag{2.15}$$

which agrees in form with the uncorrelated noise model.

For the fixed coordinate system the low elevation assumption yields

$$\underset{\sim}{R}(k) = \begin{bmatrix} r^2(k)\sigma_B^2\sin^2B(k) + \sigma_r^2\cos^2B(k) & \sin B(k)\cos B(k)[\sigma_r^2 - r(k)\sigma_B^2] & 0 \\ R_{12}=R_{21} & r^2(k)\sigma_B^2\cos^2B(k) + \sigma_r^2\sin^2B(k) & 0 \\ 0 & 0 & r^2(k)\sigma_E^2 \end{bmatrix} \quad (2.16)$$

which agrees with the uncorrelated noise model only when B(k) = 0, 90, 180, or 270 degrees. Equations (2.15) and (2.16) indicate that the uncorrelated measurement noise assumption is likely to be considerably more accurate in the rotated coordinate system than it is in the fixed cartesian coordinate system.

2. Modeling of Random Forcing Inputs

The relative magnitudes of the vector components of forcing inputs in fixed cartesian coordinates depend on the target's orientation within the coordinate system.

The track generating program described in Appendix C has been used to demonstrate the change in the relative magnitude of the x and y acceleration components of a target maneuvering in the xy plane as the bearing of the track's axis of symmetry is changed as illustrated in Fig. 2 for two bearings. Inbound trajectories with seven-g maneuvers in the xy plane were generated using the track generating program in ten separate runs with each run differing only in the bearing angle of the axis of symmetry of the trajectory.

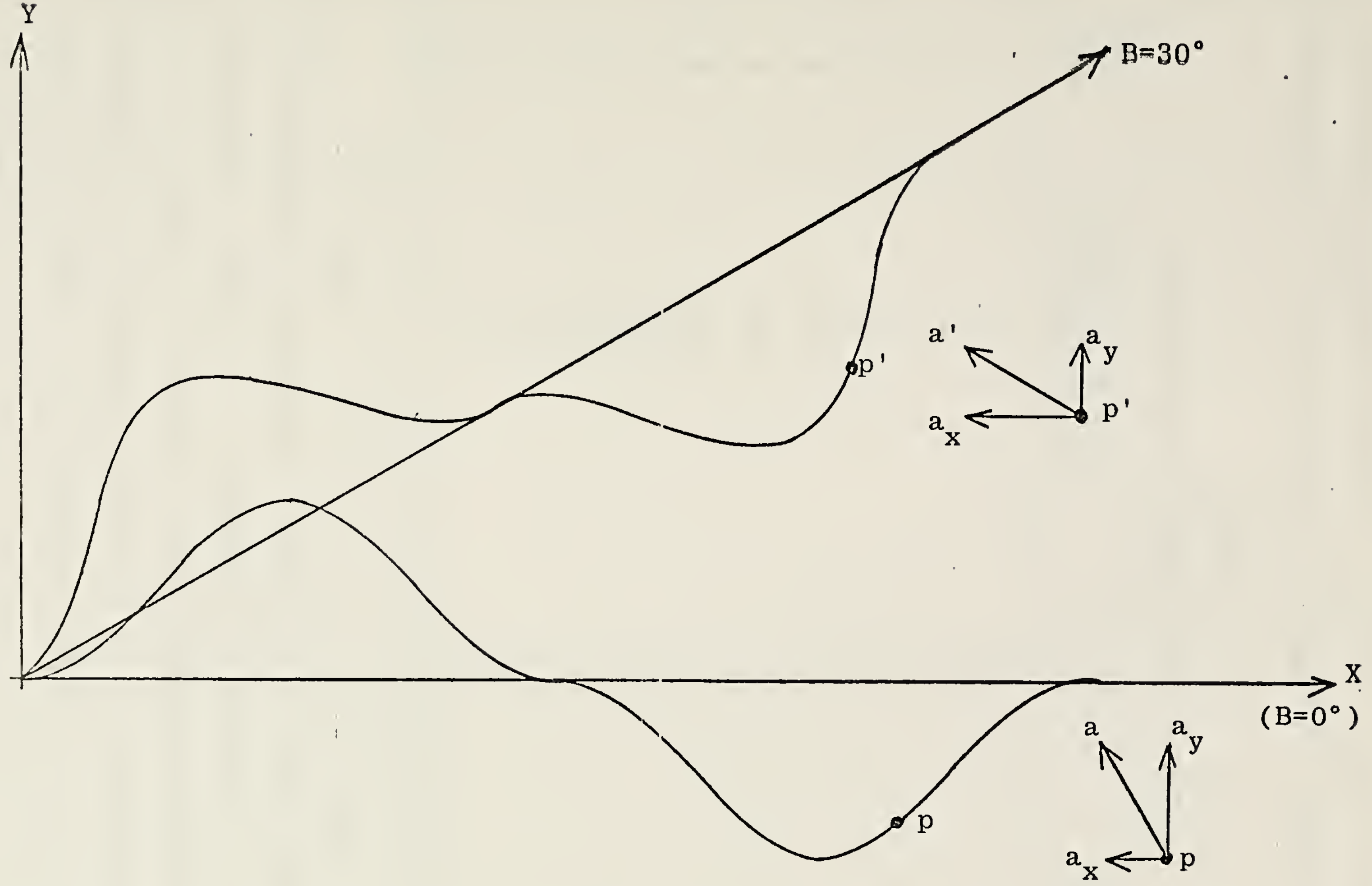


FIGURE 2 THE ROTATED XY PLANE MANEUVER SHOWING ACCELERATION COMPONENTS AT POINT P (BEFORE ROTATION) AND POINT P$'$ (AFTER ROTATION)

Table I shows the relationship between the maximum acceleration in the x-direction, $|\ddot{x}|_{max}$, and the maximum acceleration in the y-direction, $|\ddot{y}|_{max}$, for each run.

TABLE I

COMPARISON OF MAXIMUM ACCELERATIONS

RUN	BEARING OF AXIS OF SYMMETRY (deg)	$|\ddot{x}|_{max}(g)$	$|\ddot{y}|_{max}(g)$
1	0	0.6	7.0
2	10	1.4	6.9
3	20	2.6	6.7
4	30	3.6	6.1
5	40	4.7	5.4
6	50	5.5	4.6
7	60	6.2	3.5
8	70	6.8	2.5
9	80	6.9	1.4
10	90	7.0	0.7

The large variation in the relative magnitudes of the acceleration components indicates that much more accurate apriori estimates of random forcing inputs are possible if all targets approach the origin from the same bearing. Therefore a considerable advantage in filter design might be gained by rotating the coordinate system so that the x-axis is aligned with the initial detection bearing of each new target.

B. COORDINATE SYSTEM ROTATION

The radar measurements of target position are range, $r(k)$, bearing, $B(k)$, and elevation, $E(k)$ (to simplify the notation in the subsequent discussion $r(k)$, $B(k)$, and $E(k)$ are assumed to include the measurement noise). To convert the measurements to cartesian coordinates the equations are

$$x(k) = r(k)\cos B(k)\cos E(k) \tag{2.17}$$

$$y(k) = r(k)\sin B(k)\cos E(k) \tag{2.18}$$

$$z(k) = r(k)\sin E(k) \tag{2.19}$$

where once again for simplicity of notation $x(k)$, $y(k)$ and $z(k)$ are assumed to include measurement noise.

If the x-axis is aligned with the initial bearing of the target, equations (2.17) through (2.19) become

$$x'(k) = r(k)\cos \Delta B(k)\cos E(k) \tag{2.20}$$

$$y'(k) = r(k)\sin \Delta B(k)\cos E(k) \tag{2.21}$$

$$z'(k) = r(k)\sin E(k) \tag{2.22}$$

where

$$\Delta B(k) \equiv B(k) - B(0) \tag{2.23}$$

as depicted in Fig. 3.

In order to maintain the orientation of the rotated coordinate system it is necessary to store the initial bearing, $B(0)$, for use in the coordinate conversion process as indicated in equation (2.23).

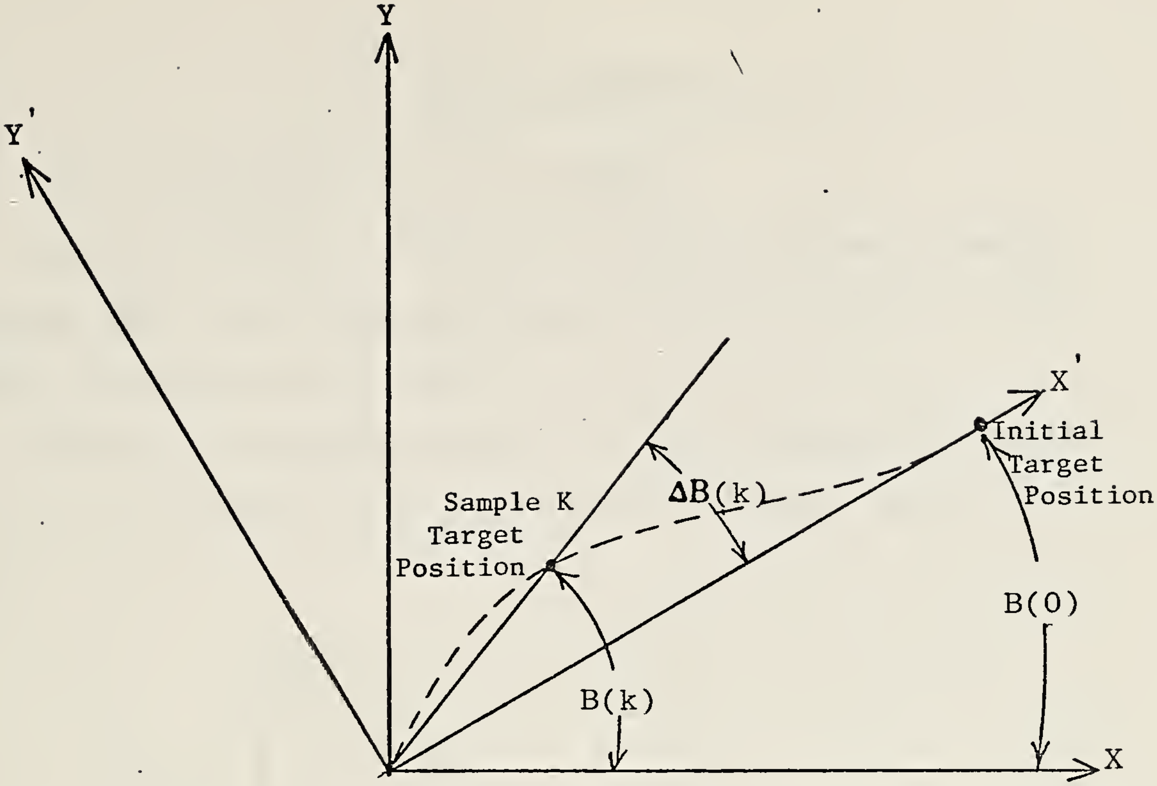


FIGURE 3 ROTATED COORDINATE SYSTEM GEOMETRY IN RELATION TO FIXED CARTESIAN COORDINATES

(Horizontal Plane Shown-Z Axis is Perpendicular to the Page)

III. FILTER DEVELOPMENT

A. THE CONSTANT-VELOCITY ($1/S^2$) MODEL

The first filter considered is based on a model which assumes that the target maintains constant velocity in all three coordinate directions.

The state transition matrix, $\underset{\sim}{\Phi}$, and $\underset{\sim}{\Gamma}$, the matrix which relates the random forcing inputs to the state vector are [1]

$$\underset{\sim}{\Phi}(T) = \left[\begin{array}{c|c|c} \begin{matrix} 1 & T \\ 0 & 1 \end{matrix} & \underset{\sim}{0} & \underset{\sim}{0} \\ \hline \underset{\sim}{0} & \begin{matrix} 1 & T \\ 0 & 1 \end{matrix} & \underset{\sim}{0} \\ \hline \underset{\sim}{0} & \underset{\sim}{0} & \begin{matrix} 1 & T \\ 0 & 1 \end{matrix} \end{array}\right] \tag{3.1}$$

and

$$\underset{\sim}{\Gamma}(T) = \left[\begin{array}{c|c|c} \begin{matrix} T^2/2 \\ T \end{matrix} & \underset{\sim}{0} & \underset{\sim}{0} \\ \hline \underset{\sim}{0} & \begin{matrix} T^2/2 \\ T \end{matrix} & \underset{\sim}{0} \\ \hline \underset{\sim}{0} & \underset{\sim}{0} & \begin{matrix} T^2/2 \\ T \end{matrix} \end{array}\right] \tag{3.2}$$

for the case of the constant-velocity system model with sampling time T.

The covariance of measurement noise matrix $\underset{\sim}{R}(k)$, for the system is given in Chapter II, equation (2.14). If the filter gains are to be calculated off-line, a model must be chosen for $\underset{\sim}{R}(k)$. This necessitates that values be chosen

for target range, target elevation, and the standard deviations of measurement noise. Range and elevation values were arbitrarily chosen as

$$r(k) = 10 \text{ kyd} \tag{3.3}$$

$$E(k) = 0 \text{ deg} \tag{3.4}$$

the measurement noise standard deviations used were

$$\sigma_r = 3 \text{ yd} \tag{3.5}$$

$$\sigma_B = \sigma_E = 0.3 \text{ mrad} \tag{3.6}$$

These values of σ_r, σ_B, and σ_E are considered to be typical of good tracking radar systems. Substituting the values given in equations (3.3) through (3.6) into equation (2.14) yields

$$\underset{\sim}{R}(k) = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 9 \end{bmatrix} \tag{3.7}$$

The system model described above is highly simplified. The model allows only for targets which fly at constant velocity. It is not expected that any target which the system might be used against would behave so well. Almost certainly there will be pilot initiated accelerations, wind buffeting, etc. which are not accounted for in the model. It is the function of the state excitation covariance matrix $\underset{\sim}{Q}$ to compensate for the inadequacies of the simplified model.

The starting point for developing the state excitation covariance matrix is the random forcing input vector, $\underset{\sim}{w}$, represented by

$$\underset{\sim}{w} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{3.8}$$

where a_x, a_y, and a_z are accelerations in the x, y, and z-coordinate directions respectively. Since any coupling between the accelerations depends on the type of random forcing experienced, it is assumed for this model that the random forcing is decoupled, i.e.

$$E\{a_x a_y\} = E\{a_x a_z\} = E\{a_y a_z\} = 0$$

The state excitation matrix $\underset{\sim}{Q}$ is defined in Chapter I as

$$\underset{\sim}{Q} = E\{\underset{\sim}{\Gamma}\underset{\sim}{w}\underset{\sim}{w}^T\underset{\sim}{\Gamma}^T\} \tag{3.9}$$

If the matrix multiplication is performed with $\underset{\sim}{\Gamma}$ and $\underset{\sim}{w}$ as previously defined, and the expectation operation performed on the product, the result is

$$\underset{\sim}{Q} = \left[\begin{array}{c|c|c} \begin{matrix} \frac{a_x^2T^4}{4} & \frac{a_x^2T^3}{2} \\ \frac{a_x^2T^3}{2} & a_x^2T^2 \end{matrix} & \underset{\sim}{0} & \underset{\sim}{0} \\ \hline \underset{\sim}{0} & \begin{matrix} \frac{a_y^2T^4}{4} & \frac{a_y^2T^3}{2} \\ \frac{a_y^2T^3}{2} & a_y^2T^2 \end{matrix} & \underset{\sim}{0} \\ \hline \underset{\sim}{0} & \underset{\sim}{0} & \begin{matrix} \frac{a_z^2T^4}{4} & \frac{a_z^2T^3}{2} \\ \frac{a_z^2T^3}{2} & a_z^2T^2 \end{matrix} \end{array}\right] \tag{3.10}$$

The values assumed for a_x, a_y, and a_z for off-line calculation of the filter gains have a significant effect on the filter's performance. Much of the simulation work reported in Chapter IV is devoted to the study of the effects of different assumed values of a_x, a_y, and a_z on the filter's performance.

The Kalman filter requires that an apriori estimate of the system state vector be made. The initial state estimate vector used in all simulation runs is $\underset{\sim}{x}(0/-1) = \underset{\sim}{0}$, based on the reasoning that in a fixed cartesian coordinate system this is the mean of all possible initial state vectors. Since the system being simulated assumes a rotated cartesian coordinate system as described in Chapter II, it might be argued that the x coordinate position could be estimated as some anticipated detection range, and the x-coordinate velocity as a typical aircraft speed in the negative x-direction. However, since the filter converges rapidly as it is presently initialized, it is felt that to revise the initialization procedure would only serve to complicate the model and would provide no foreseeable benefit.

What is believed to be the simplest approach is taken again in choosing the initial covariance of estimation error matrix. It is desired that the initial position measurements be used as the initial position state estimates in order to have the filter track as quickly as possible. This end is accomplished quite simply by making the diagonal elements of the initial covariance of estimation error matrix

very large (e.g. 10^9). The effect of doing this can be seen by examining the Kalman filter gain equation

$$\underset{\sim}{G}(k) = \underset{\sim}{P}(k/k-1)\underset{\sim}{H}^T[\underset{\sim}{H}\underset{\sim}{P}(k/k-1)\underset{\sim}{H}^T+\underset{\sim}{R}(k)]^{-1} \tag{3.11}$$

and considering the scalar case

$$\begin{aligned} G(k) &= P(k/k-1)H[HP(k/k-1)H+R(k)]^{-1} \\ &= \frac{P(k/k-1)H}{H^2P(k/k-1)+R(k)} \end{aligned} \tag{3.12}$$

where since $\underset{\sim}{H}$ is the observation matrix the scalar equivalent would simply be unity. If P(k/k-1) is made very large with respect to R(k) for k=0, then we have

$$G(0) = \frac{P(0/-1)}{P(0/-1)+R(0)} \simeq 1 \tag{3.13}$$

which makes the first estimate equal the first measurement -- a logical result in the presence of a high degree of uncertainty.

The initial covariance of estimation error matrix used for all simulations is

$$\underset{\sim}{P}(0/-1) = \begin{bmatrix} 10^9 & & & & \underset{\sim}{0} \\ & 10^9 & & & \\ & & \ddots & & \\ & & & 10^9 & \\ & & & & \ddots \\ \underset{\sim}{0} & & & & 10^9 \end{bmatrix} \tag{3.14}$$

B. THE CORRELATED RANDOM ACCELERATION MODEL

One approach used to simulate the effects of target motion involves introducing correlated random acceleration inputs to a $1/S^2$ model. The correlated random accelerations are generated by an input of white noise into a coloring filter having a transfer function of $\alpha/(S+\alpha)$. The value of α can be selected to vary the time constant of the maneuver. This results in an overall transfer function for the model of target motion of

$$T(S) = \frac{\alpha}{S^2(S+\alpha)} \tag{3.15}$$

The 9×1 system state vector assumed for this model of target motion is

$$\underset{\sim}{x} = \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \ddot{x}(t) \\ y(t) \\ \dot{y}(t) \\ \ddot{y}(t) \\ z(t) \\ \dot{z}(t) \\ \ddot{z}(t) \end{bmatrix} \tag{3.16}$$

McKinley [4] describes this model in detail and gives the state difference equations (for one dimension) from which the state transition matrix

$$\underset{\sim}{\Phi}_i(T) = \begin{bmatrix} 1 & T & \frac{T}{\alpha} - \frac{1}{\alpha^2}\,[1-e^{-\alpha T}] \\ 0 & 1 & \frac{1}{\alpha}\,[1-e^{-\alpha T}] \\ 0 & 0 & e^{-\alpha T} \end{bmatrix} \tag{3.17}$$

and

$$\underset{\sim}{\Gamma}_i(T) = \begin{bmatrix} \frac{T^2}{2} - \frac{T}{\alpha} + \frac{1}{\alpha^2}\,[1-e^{-\alpha T}] \\ T - \frac{1}{\alpha}\,[1-e^{-\alpha T}] \\ 1 - e^{-\alpha T} \end{bmatrix} \tag{3.18}$$

can be extracted.

The matrices $\underset{\sim}{\Phi}_i(T)$ and $\underset{\sim}{\Gamma}_i(T)$ apply to a three-state system in one coordinate direction. For the full, nine-state three-coordinate system, the 9×9 matrix $\underset{\sim}{\Phi}(T)$ is

$$\underset{\sim}{\Phi}(T) = \begin{bmatrix} \underset{\sim}{\Phi}_x(T) & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\Phi}_y(T) & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{\Phi}_z(T) \end{bmatrix} \tag{3.19}$$

where $\underset{\sim}{\Phi}_x(T)$, $\underset{\sim}{\Phi}_y(T)$, and $\underset{\sim}{\Phi}_z(T)$ are each equal to the 3×3 matrix $\underset{\sim}{\Phi}_i(T)$, and the 9×3 matrix $\underset{\sim}{\Gamma}(T)$ is

$$\underset{\sim}{\Gamma}(T) = \begin{bmatrix} \underset{\sim}{\Gamma}_x(T) & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\Gamma}_y(T) & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{\Gamma}_z(T) \end{bmatrix} \tag{3.20}$$

where $\underset{\sim}{\Gamma}_x(T)$, $\underset{\sim}{\Gamma}_y(T)$ and $\underset{\sim}{\Gamma}_z(T)$ are each equal to the 3×1 vector $\underset{\sim}{\Gamma}_i(T)$.

In order to make the derivation of the $\underset{\sim}{Q}$ matrix less difficult to follow the following notation is adopted.

$$\underset{\sim}{\Gamma}(T) = \begin{bmatrix} \gamma_1 & 0 & 0 \\ \gamma_2 & 0 & 0 \\ \gamma_3 & 0 & 0 \\ 0 & \gamma_1 & 0 \\ 0 & \gamma_2 & 0 \\ 0 & \gamma_3 & 0 \\ 0 & 0 & \gamma_1 \\ 0 & 0 & \gamma_2 \\ 0 & 0 & \gamma_3 \end{bmatrix} \tag{3.21}$$

where

$$\gamma_1 = \frac{T^2}{2} - \frac{T}{\alpha} + \frac{1}{\alpha^2}\,[1-e^{-\alpha T}] \ , \tag{3.22}$$

$$\gamma_2 = T - \frac{1}{\alpha}\,[1-e^{-\alpha T}] \ , \text{ and} \tag{3.23}$$

$$\gamma_3 = 1 - e^{-\alpha T} \quad . \tag{3.24}$$

The forcing input vector $\underset{\sim}{w}$ and the state excitation covariance matrix $\underset{\sim}{Q}$ are defined as in equations (3.8) and (3.9), respectively. If the matrix multiplication is done and the expectation operation performed on the product, the result is

$$\underset{\sim}{Q} = \begin{bmatrix} a_x^2 \ \underset{\sim}{q} & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & a_y^2 \ \underset{\sim}{q} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & a_z^2 \ \underset{\sim}{q} \end{bmatrix} \tag{3.25}$$

where

$$\underset{\sim}{q} = \begin{bmatrix} \gamma_1^2 & \gamma_1\gamma_2 & \gamma_1\gamma_3 \\ \gamma_1\gamma_2 & \gamma_2^2 & \gamma_2\gamma_3 \\ \gamma_1\gamma_3 & \gamma_2\gamma_3 & \gamma_3^2 \end{bmatrix} \tag{3.26}$$

The assumed covariance of measurement noise matrix, $\underset{\sim}{R}$, is the same as given in equation (3.7) for the constant-velocity system.

The constant α is inversely proportional to the assumed maneuver time constant. For example, the value of $\alpha=0.5$ corresponds to a time constant of two seconds which is considered to be typical for an aircraft evasive maneuver. Different time constants could conceivably be applied to each coordinate direction, however, in this thesis one value of α is applied throughout the system model.

The $\alpha/S^2(S+\alpha)$ model was simulated using values of $\alpha=0.1$, 0.3, 0.5 and 1.0. Based on the results of the simulations -- reported in Chapter IV -- $\alpha=0.5$ was chosen as the value providing the best overall performance in the limited number of tests conducted. This does not suggest that $\alpha=0.5$ is an optimum choice.

IV. MONTE-CARLO SIMULATION

The Monte-Carlo simulation program described in Appendix A was used to evaluate variations of the two basic models described in Chapter III.

A. TEST TRAJECTORIES

A set of four trajectories was chosen as a standard for use in simulating all models in order to facilitate comparison of the simulation results. All of the trajectories are products of the track generating program described in Appendix C. Each track is comprised of 60 state trajectory data samples with a sampling rate of four hertz.

The tracks are developed so that the initial target bearing coincides with the x axis to simulate the rotated coordinate system. The addition of measurement noise will cause the initial bearing to not coincide exactly with the x axis, however, no significant error is expected.

1. The Short-Range, Constant-Velocity Trajectory

The short-range, constant-velocity trajectory is a straight, inbound trajectory, beginning at a range of 3,850 yards at 30 degrees elevation and proceeding toward the origin with a range rate of 230 yd/s to a final range of 460 yards.

2. The Long-Range, Constant-Velocity Trajectory

The long-range, constant-velocity trajectory is a straight inbound track beginning at a range of 9,225 yards

at 30 degrees elevation and proceeding toward the origin with a range rate of 230 yd/s to a final range of 5,817 yards.

3. The Short-Range Trajectory With Maneuver

The short-range trajectory with a maneuver, like the short-range, constant-velocity track, begins at a range of 3,850 yards at 30 degrees elevation and proceeds toward the origin with a target speed of 230 yd/s. Commencing at a range of 3,500 yards the target undergoes a two-g, $\sin^2$ maneuver of 13.5 seconds duration. The maneuver is completed at the final sample range of 470 yards.

4. The Long-Range Trajectory With Maneuver

The long-range trajectory with a maneuver begins at a range of 9,225 yards at 30 degrees elevation. The target proceeds toward the origin with a speed of 230 yd/s. Commencing at a range of 8,900 yards the target undergoes a two-g, $\sin^2$ maneuver of 13.5 seconds duration. The maneuver is completed at the final sample range of 5,853 yards.

5. Other Trajectories

Representative variations of the two basic models were simulated using various trajectories in addition to the four standard trajectories described above in order to insure that the characteristics shown in the simulations were valid for trajectories other than the standard set.

The additional trajectories used include a constant-velocity crossing trajectory which has its initial point on the x-axis (simulating the rotated coordinate system) at a

range of 3,500 yards and proceeds to diverge from the x-axis at an angle of ten degrees as range decreases, and inbound trajectories with two-g and 2.5-g $\sin^2$ maneuvers of ten seconds duration occurring at various ranges.

B. ACCELERATIONS ASSUMED

The variations of the two basic system models differ primarily in the values of accelerations assumed for the components of the forcing input vector, $\underset{\sim}{w}(k)$, which is used in calculating the state excitation covariance matrix $\underset{\sim}{Q}(k)$. As indicated in equation (3.10) for the constant-velocity model, and equation (3.25) for the correlated random acceleration model, larger assumed values of acceleration result in larger corresponding entries in the $\underset{\sim}{Q}$ matrix. Solution of equations (1.7) through (1.9) shows that a consequence of larger $\underset{\sim}{Q}$ matrix entries is larger entries in the gain matrix $\underset{\sim}{G}(k)$. This results in the filter being more responsive to target maneuvers, but it is also more susceptible to estimation error caused by measurement noise. The assumption of different values for the components of $\underset{\sim}{w}(k)$ is done in an effort to determine the $\underset{\sim}{Q}$ matrix which best serves to compensate for modeling inadequacies as discussed in Chapter III.

The values of acceleration assumed are chosen as being representative of manned aircraft accelerations in the rotated cartesian coordinate system. It is predicted that an attacking aircraft during evasive maneuvering would have accelerations in the y and z-directions of three g's or less

and accelerations in the x-direction of less than one g. The values used for the simulation runs are listed in Tables II through IV along with the corresponding simulation prediction accuracy statistics.

C. SIMULATION RESULTS

The main criterion chosen for evaluation of model performance is the position-state prediction accuracy as discussed in Appendix B, although the means and covariances of estimation error are also calculated. The predictions for each model are calculated using the state transition matrix for the model as developed in Chapter III.

The statistics tabulated for comparison are the percentage of predictions per run which were in error by less than 24 feet -- labeled "hit %" -- and the percentage of predictions per run in error by less than 48 feet -- labeled "scare %".

The results of the various constant-velocity model simulations are listed in Table II. All statistics listed are products of 100 member Monte-Carlo ensembles.

The correlated-random-acceleration model was tested for several values of α as well as various assumed accelerations. The simulation results for $\alpha=0.3$ and $\alpha=0.5$ are listed in Tables III and IV respectively. All other values of α tested provided inferior performance and the results are not tabulated.

The simulation results listed in Tables II through IV are considered to be representative of all prediction

TABLE II CONSTANT-VELOCITY MODEL SIMULATION RESULTS

ASSUMED ACCELERATIONS (g)	CONSTANT-VELOCITY TRAJECTORIES				MANEUVERING TRAJECTORIES			
	Short-Range		Long-Range		Short-Range		Long-Range	
	Hit%	Scare %	Hit %	Scare %	Hit %	Scare %	Hit %	Scare %
$a_x=0$ $a_y=0$ $a_z=0$	86.4	91.4	34.7	59.4	5.9	9.14	0.0	0.095
$a_x=0.1$ $a_y=0.1$ $a_z=0.1$	86.4	91.4	32.4	59.2	5.9	11.5	0.0	0.095
$a_x=0.1$ $a_y=0.2$ $a_z=0.2$	88.4	91.4	25.2	58.3	13.0	27.4	0.0	0.67
$a_x=0.1$ $a_y=1.0$ $a_z=1.0$	81.4	91.2	4.62	21.57	15.2	28.5	0.33	1.10
$a_x=0.1$ $a_y=2.0$ $a_z=2.0$	74.2	90.8	2.0	9.1	13.8	31.3	0.14	0.86
$a_x=0.1$ $a_y=3.0$ $a_z=3.0$	63.4	86.1	1.1	6.1	13.8	32.4	0.29	0.95

TABLE III CORRELATED-RANDOM-ACCELERATION MODEL WITH α=0.3

ASSUMED ACCELERATIONS (g)	CONSTANT-VELOCITY TRAJECTORIES				MANEUVERING TRAJECTORIES			
	Short-Range		Long-Range		Short-Range		Long-Range	
	Hit %	Scare %	Hit %	Scare %	Hit %	Scare %	Hit %	Scare %
a_x=0.1 a_y=0.1 a_z=0.1	71.3	79.1	2.9	15.1	5.0	18.6	0.0	0.4
a_x=0.1 a_y=0.2 a_z=0.2	71.3	79.1	3.0	15.1	13.8	22.6	0.1	0.3
a_x=0.1 a_y=1.0 a_z=1.0	70.6	78.9	1.5	8.3	8.5	28.7	0.1	0.4
a_x=0.1 a_y=2.0 a_z=2.0	66.7	78.6	0.8	3.9	8.6	20.1	0.0	0.1
a_x=0.1 a_y=3.0 a_z=3.0	62.9	77.7	0.4	2.1	6.4	21.7	0.0	0.1

TABLE IV CORRELATED-RANDOM-ACCELERATION MODEL WITH $\alpha=0.5$

ASSUMED ACCELERATIONS (g)	CONSTANT-VELOCITY TRAJECTORIES				MANEUVERING TRAJECTORIES			
	Short-Range		Long-Range		Short-Range		Long-Range	
	Hit %	Scare %	Hit %	Scare %	Hit %	Scare %	Hit %	Scare %
$a_x=0.1$ $a_y=0.5$ $a_z=0.5$	73.4	81.1	6.6	24.0	4.6	18.8	not tested	
$a_x=0.1$ $a_y=1.0$ $a_z=1.0$	73.4	81.1	6.0	23.8	21.2	30.2	not tested	
$a_x=0.2$ $a_y=1.0$ $a_z=1.0$	72.8	81.1	4.6	20.5	20.4	30.2	not tested	
$a_x=0.1$ $a_y=2.0$ $a_z=2.0$	73.1	81.1	4.1	18.7	17.3	29.3	not tested	
$a_x=0.2$ $a_y=2.0$ $a_z=2.0$	72.3	81.1	3.2	16.7	16.5	29.1	not tested	

accuracy statistics obtained, with the exception that Table II does not present statistics indicating the effect on filter performance when the assumed x-direction acceleration a_x, is varied. It was determined early in the simulation phase that the constant-velocity model performs best against the test trajectories used when a_x is approximately 0.1 g, therefore in the interest of computer time conservation only this value of a_x was used for the final set of simulation runs.

Also in order to conserve computer time the correlated-random-acceleration model with $\alpha=0.5$ was not tested against the long-range trajectory with a maneuver. This is considered a justifiable omission since no appreciable improvement or further degradation of prediction accuracy in this area is anticipated.

In Table V the five best performing filters against each test trajectory are ranked according to the prediction accuracy statistics shown in Table II through IV. The "hit" and "scare" percentages for each ranked filter are included with the model description. For the long-range, maneuvering target only two of the estimators tested produced "scare" percentages of one percent -- the filter ranked highest is chosen because it produced smaller average miss distances than the second and third ranked filters for 17 of 21 predictions per run even though it scored only two "scares" in the 100-member ensemble.

TABLE V FILTER PERFORMANCE RANKINGS

SHORT-RANGE CONSTANT-VELOCITY TRAJECTORY	LONG-RANGE CONSTANT-VELOCITY TRAJECTORY	SHORT-RANGE MANEUVERING TRAJECTORY	LONG-RANGE MANEUVERING TRAJECTORY
(1)$1/S^2$ Model $a_x=0.1g$ $a_y=a_z=0.2g$ (88.4/91.4)	(1)$1/S^2$ Model $a_x=a_y=a_z=0$ (34.7/59.2)	(1)$0.5/S^2(S+0.5)$Model $a_x=0.1g$ $a_y=a_z=1.0g$ (21.2/30/2)	(1)$1/S^2$ Model $a_x=a_y=a_z=0.1g$ (0.0/0.1) Lower Average Miss Distance for 17 of 21 predictions per run
(2)$1/S^2$ Model $a_x=a_y=a_z=0.1g$ or $a_x=a_y=a_z=0.0$ (86.4/92.4)	(2)$1/S^2$ Model $a_x=a_y=a_z=0.1g$ (32.4/59.2)	(2)$0.5/S^2(S+0.5)$Model $a_x=0.2g$ $a_y=a_z=1.0g$ (20.4/32.2)	(2)$1/S^2$ Model $a_x=0.1g$ $a_y=a_z=1.0g$ (0.3/1.1)
(3)$1/S^2$ Model $a_x=0.1g$ $a_y=a_z=1.0g$ (81.4/91.2)	(3)$1/S^2$ Model $a_x=0.1g$ $a_y=a_z=0.2g$ (24.2/58.3)	(3)$0.5/S^2(S+0.5)$Model $a_x=0.1g$ $a_z=a_z=2.0g$ (17.3/29.3)	(3)$1/S^2$ Model $a_x=0.1g$ $a_y=a_z=3.0g$ (0.1/1.0)
(4)$1/S^2$ Model $a_x=0.1g$ $a_y=a_z=2.0g$ (74.2/90.8)	(4)$0.5/S^2(S+0.5)$Model $a_x=0.1g$ $a_y=a_z=0.5g$ (6.6/24.0)	(4)$1/S^2$ Model $a_x=0.1g$ $a_y=a_z=3.0g$ (13.8/32.4)	
(5)$0.5/S^2(S+0.5)$Model $a_x=0.1g$ $a_y=a_z=1.0g$ (73.4/81.1)	(5)$0.5/S^2(S+0.5)$Model $a_x=0.1g$ $a_y=a_z=1.0g$ (6.0/23.8)	(5)$1/S^2$ Model $a_x=0.1g$ $a_y=a_z=2.0g$ (13.8/31.3)	

D. COMPARISON OF PREDICTION ACCURACY AND ESTIMATION ERROR STATISTICS

An effort to correlate the prediction accuracy statistics with the estimation error means and variances for the sample estimate from which the prediction is made indicates that for constant-velocity targets, where the estimation error means are small for all the models, the model with the smaller variances of estimation error -- especially for the velocity state estimates -- can be expected to produce more consistently accurate predictions. For example, a comparison of the statistics for two $1/S^2$ models, one with forcing inputs assumed to be zero and the other with assumed values of 0.1 g in the x-direction and two g's in the y and z-directions, shows that both filters in general have small means of estimation error. The estimate at sample number 18, $\underset{\sim}{\hat{x}}(18/18)$, upon which the prediction of target position at the time of sample number 53, $\underset{\sim}{\hat{x}}(53/18)$, is based in the long-range, constant-velocity trajectory, has small means of estimation error for both filters (less than one yard for position states and one yd/s for velocity states). The variances of y and z-direction velocity estimation error for the model which assumes two-g acceleration in the y and z directions are much larger (an order of magnitude larger for the y and z-direction velocity states). This model scores 18 "scares" at sample number 53 over the 100 run ensemble while the model which assumes zero forcing inputs scores 100

"scares". The trend indicated in this example holds for all constant-velocity trajectory statistics tested.

A similar analysis of the statistics generated for short-range maneuvering targets, where the means of estimation error are in some cases more than 20 yards for the position states and more than 30 yd/s for the velocity states, indicates that the better prediction accuracy statistics are produced by the model with the lower means of estimation error. The variance of estimation error does not appear to be a significant factor at short range in any of the models tested. Notice that the mean estimation errors are much larger than the measurement errors, this is due to the target maneuver occurring after the constant-velocity filter has reached steady state.

At long ranges, any "hits" scored against a maneuvering target are considered to be either purely by chance or due to the type of maneuver which the target executes (i.e. the target may pass close to the predicted position even though it has maneuvered radically during the projectile time-of-flight).

The average prediction miss distance is considered to be the best prediction accuracy statistic for use in evaluating a filter's performance against a long-range, maneuvering target. The model ranked first against long-range maneuvering targets in Table V has larger means and smaller variances of estimation error than the second and third ranked models. These statistics indicate that the estimator is not

tracking the target accurately, but is generating predictions of future target position in the right general area. This is suspected to be largely because the target maneuvers about an axis of symmetry in the test trajectory rather than at random.

V. ADAPTIVE FILTERS

A. THE SWITCH-ON-RANGE ADAPTIVE FILTER

The simpler of the two adaptive filtering methods proposed is the switch-on-range adaptive filter. This scheme is based on the idea that since the measurement noise in cartesian coordinates is proportional to the target range, relatively high filter gains can be used on a short-range target without significant degradation of the estimate quality for a constant-velocity target and with significant improvement of the estimate quality for a maneuvering target.

Upon target detection the estimator uses a gain schedule which is based on a small forcing inputs -- such as 0.1 g in each coordinate direction. This "low Q" gain schedule is continued until the target range is less than a predetermined threshold. Upon reaching the switching threshold range the estimator switches to a steady-state gain matrix which is based on relatively large forcing inputs ("high Q").

The $1/S^2$ model is used for both the "low Q" and "high Q" filters. This model was chosen because of its superior performance against constant-velocity targets, especially at long range, as indicated in the Monte-Carlo simulation results of Chapter IV.

The "high Q" filter uses a steady-state gain matrix because it is assumed that a good estimate is available at the time the switching occurs.

The switching threshold range for simulation purposes is chosen as 3,333 yards. The simulation results of Chapter IV indicate that the "high Q" filters based on the $1/S^2$ model with assumed accelerations of one to three g's in the y and z-directions perform relatively well within this range against either maneuvering or constant-velocity targets.

The "high Q" gain matrix used in the simulation is the steady-state gain matrix of the $1/S^2$ model with assumed accelerations of 0.1 g in the x-direction and three g's in the y and z-directions. The gains are computed from the model developed in Chapter III.

A FORTRAN listing of the switch-on-range adaptive filter subroutine is included in Appendix D.

B. SWITCH-ON-RESIDUALS ADAPTIVE FILTER #1

The switch-on-residuals adaptive filter is more sophisticated and more complex than the switch-on-range filter previously described.

The first version of the switch-on-residuals adaptive filter is designed to track with a "low Q" gain schedule based on the constant-velocity model until a component of the filter residual vector

$$\underset{\sim}{\rho}(k) = \underset{\sim}{Z}(k)-\underset{\sim}{H}\hat{\underset{\sim}{x}}(k/k-1) \qquad (5.1)$$

exceeds a calculated gating level,

$$\ell_i(k) = 2\sqrt{P_{ii}(k/k)+R_{ii}(k)}\ , \quad i=1,2,3, \qquad (5.2)$$

on two successive samples.

$P_{ii}(k/k)$ is the theoretical variance of position estimate error and $R_{ii}(k)$ is the theoretical variance of measurement noise in one coordinate direction.

The values used for the theoretical variance of position estimate error are taken from the steady-state theoretical covariance of estimation error matrix for the model from which the "high Q" gains are calculated.

The values used for the theoretical variance of measurement noise in the gating level calculation are calculated on-line from the diagonal terms of the covariance of measurement noise matrix given in equation (2.14) and vary with the target location.

If the occurrence of a residual value greater than the gating level at time i is defined as event A_i and the probability of this occurring on the next sample is event A_{i+1}, A_i and A_{i+1} are independent events. The probability of two successive samples with residuals greater than the gating level is

$$P(A_i A_{i+1}) = P(A_i)P(A_{i+1}) \ . \qquad (5.3)$$

If only the $R_{ii}(k)$ term of the equation (5.2) is considered, then the gating level is equal to twice the standard deviation of measurement noise. For a normal distribution the probability of an event differing from the mean by more than two standard deviations is given as

$$P(A_i) = P(A_{i+1}) = 0.0228 \ .$$

Substitution of this value into equation (5.3) yields a probability of

$$P(A_i A_{i+1}) = (0.0228)^2 = 5.20 \times 10^{-4}$$

which is the probability that the residual is greater than the gating level due to estimation error or measurement noise when no target maneuver is experienced.

The filter is independently adaptive in each coordinate direction. For example, when $\rho_1(k)$ is greater than $\ell_1(k)$ for two successive samples, the x-direction states are re-initialized by accepting the latest measurement as the position estimate and beginning at k=0 with a "high Q" gain schedule based on the $\alpha/S^2(S+\alpha)$ system model with values of and assumed forcing inputs chosen to provide good performance against a maneuvering target.

The y-direction and z-direction state estimates continue to be based on the constant-velocity ($1/S^2$) model. The $\alpha/S^2(S+\alpha)$ filter gains continue to be used in making the x-direction estimates until three successive samples have residuals less than one-half the gating level.

Then the inequality

$$\rho_1(k) < \sqrt{P_{11}(k/k)+R_{11}(k)} \qquad (5.4)$$

is satisfied for three successive samples the filter switches back to the "low Q" gain schedule assuming the same time index as would have been used for the current sample if no adaptive switching had occurred.

For the Monte-Carlo simulation of the switch-on-residuals filter, the "low Q" gains are based on the $1/S^2$ model with assumed forcing inputs of 0.1 g in each coordinate direction since it is proposed that those gains will be used exclusively against constant-velocity targets. This choice is based on the simulation results listed in Chapter IV.

Also based on the simulation results listed in Chapter IV, the model chosen for calculation of the "high Q" gains is the $\alpha/S^2(S+\alpha)$ model with $\alpha=0.5$ and assumed forcing inputs of 0.1 g in the x-direction and one g in the y and z-directions.

C. SWITCH-ON-RESIDUALS ADAPTIVE FILTER #2

The second version of the switch-on-residuals adaptive filter differs from the first in the manner in which the switching gates are calculated.

In this version the gating level is calculated as

$$\ell_i(k) = 3\sqrt{P_{ii}(k/k)+R_{ii}(k)}\ , \quad i=1,2,3 \qquad (5.5)$$

and the switch is made to the "high Q" gain schedule when the gating level is exceeded by one sample residual.

The switch back to the "low Q" gain schedule occurs when the residual is less than $0.88\ \ell_i(k)$ for two successive samples (the coefficient 0.88 is an arbitrary choice for a value greater than 0.5 but less than one). In this version when the filter switches back to the "low Q" gains the index for the "high Q" gain schedule is not reset immediately but is delayed one sample so that if the residual exceeds the

gating level on the first sample after the filter returns to the "low Q" gain schedule, the filter does not reinitialize by accepting the position measurement as the estimate but simply switches back to the "high Q" gain schedule using the same time index as would have been used if the switch to the "low Q" gain schedule had not taken place.

The gain schedules used in this filter are the same as those used in the first version of the switch-on-residuals adaptive filter described in the previous section.

D. THE COMPROMISE FILTER

The compromise filter is an attempt to pick one gain schedule which provides adequate performance for all types of trajectories without adaptive switching. The filter simulation results listed in Chapter IV indicate no clear choice for the compromise filter.

Because of its good performance against constant-velocity targets at all ranges and its acceptable performance against short-range maneuvering targets, the $1/S^2$ model which assumes accelerations of 0.1 g in the x-direction and 0.2 g in the y and z-directions is chosen for calculation of the compromise filter gains.

E. SUMMARY OF ADAPTIVE FILTER DEVELOPMENT

Four adaptive filters have been developed for evaluation by Monte-Carlo simulation, Table VI provides a summary of their key features.

TABLE VI SUMMARY OF ADAPTIVE FILTERS

ADAPTIVE FILTER	LOW Q FILTER MODEL	HIGH Q FILTER MODEL	SWITCH TO HIGH Q FILTER	SWITCH TO LOW Q FILTER
Switch-on-Range	$1/S^2$ Model $a_x=a_y=a_z=0.1g$	$1/S^2$ Model $a_x=0.1g$ $a_y=a_z=3g$	Range < 3333 yd	Range $\geq$ 3333 yd
Switch-on Residuals #1	$1/S^2$ Model $a_x=a_y=a_z=0.1g$	$\alpha/S^2(S+\alpha)$Model $\alpha=0.5$ $a_x=0.1g$ $a_y=a_z=1.0g$	$\rho_i>2\sqrt{P_{ii}(k/k)+R_{ii}(k)}$ for 2 successive samples	$\rho_i<1/2\sqrt{P_{ii}(k/k)+R_{ii}(k)}$ for 3 successive samples
Switch-on Residuals #2	$1/S^2$ Model $a_x=a_y=a_z=0.1g$	$\alpha/S^2(S+\alpha)$Model $\alpha=0.5$ $a_x=0.1g$ $a_y=a_z=1.0g$	$\rho_i>3\sqrt{P_{ii}(k/k)+R_{ii}(k)}$ for one sample	$\rho_i<0.88\sqrt{P_{ii}(k/k)+R_{ii}(k)}$ for 2 successive samples

VI. ADAPTIVE FILTER SIMULATION

The Monte-Carlo simulation program described in Appendix A is used in conjunction with the adaptive filter subroutines listed in Appendix D to evaluate the performance of the three adaptive filter schemes and the compromise non-adaptive filter proposed in Chapter V.

A. TEST TRAJECTORIES

A set of six test trajectories was used for the simulation of each of the adaptive filter configurations and the compromise filter. Each test trajectory is a product of the track generating program described in Appendix C and is comprised of 160 state trajectory data samples with a four-hertz sampling rate. All six of the test trajectories represent a target proceeding inbound from a range of 8,000 yards and an elevation of 20 degrees at a speed of 200 yd/s with a dive angle of 20 degrees. The trajectories differ in the target maneuvers incorporated:

Trajectory No. 1 - No maneuver

Trajectory No. 2 - A two-g maneuver of 13.5 seconds duration commencing at a range of 4,000 yards.

Trajectory No. 3 - A two-g maneuver of 13.5 seconds duration commencing at a range of 2,700 yards.

Trajectory No. 4 - A two-g maneuver of 13.5 seconds duration commencing at a range of 3,333 yards.

Trajectory No. 5 - A 2.5-g maneuver of 10 seconds duration commencing at a range of 2,700 yards.

Trajectory No. 6 - A 2.5-g maneuver of 10 seconds duration commencing at a range of 4,000 yards.

B. SIMULATION RESULTS

The prediction accuracy statistics produced by the Monte-Carlo simulation of the switch-on-range and switch-on-residuals adaptive filters and the compromise constant-velocity filter are listed in Table VII. The statistics presented are products of 80 member ensembles.

The adaptive filter predictions are generated using the constant-velocity model state transition matrix for all configurations. A small number of simulation runs made using the correlated-random-acceleration model state transition matrix produced slightly lower "hit" and "scare" percentages.

TABLE VII MONTE-CARLO SIMULATION RESULTS FOR ADAPTIVE FILTERS

TEST TRAJECTORY NO.	SWITCH-ON-RANGE		#1 SWITCH-ON-RESIDUALS		#2 SWITCH-ON-RESIDUALS		COMPROMISE CONSTANT-VELOCITY	
	Hit %	Scare %	Hit %	Scare %	Hit %	Scare %	Hit %	Scare %
1	81.8	93.4	74.4	81.4	86.6	92.6	85.7	93.8
2	47.2	57.8	33.8	45.4	44.2	60.0	40.4	56.9
3	60.1	75.3	47.9	60.2	58.3	72.0	55.2	69.8
4	52.3	65.7	39.7	53.2	50.4	63.3	45.1	65.1
5	60.2	75.2	45.5	60.7	59.6	73.0	56.7	70.8
6	54.7	67.8	41.8	53.8	51.9	65.9	45.5	64.2

VII. SUMMARY AND CONCLUSIONS

A. TEST RESULT SUMMARY

1. Simulation Result Summary for the Two Basic Models

The simulation results presented in Chapter IV show that the constant-velocity ($1/S^2$) model with relatively small values of assumed forcing inputs provides the better performance against constant-velocity targets at both short range and long range. By increasing the value assumed for the forcing inputs to the $1/S^2$ model its performance against maneuvering targets can be improved but there is a corresponding loss in the performance for constant-velocity targets, particularly at long range where the transverse measurement noise components are relatively large.

The correlated-random-acceleration model provides better performance against maneuvering targets and reasonably good performance against constant-velocity targets at short range, but its performance against long-range, constant-velocity targets, as measured by the prediction accuracy statistics listed in Chapter IV, is inferior to the "low Q" $1/S^2$ models.

None of the models tested provided good prediction accuracy against the long-range maneuvering target. This is to be expected because of the long time of flight over which the prediction must be made. Even a perfect estimator would surely provide inaccurate predictions when the predictions must be made ten seconds ahead on a target which maneuvers

in the interim. Average miss distance was chosen as the most appropriate statistic for evaluating a filter's performance against a long-range, maneuvering target.

2. Adaptive Filter Simulation Summary

The simulation results for the adaptive filters and the compromise filter listed in Chapter VI indicate that the switch-on-range adaptive filter provides the best overall performance against the six targets trajectories included in the test. Although the performance of the second version of the switch-on-residuals adaptive filter is comparable.

It was anticipated that the more sophisticated switch-on-residuals adaptive filter would provide the better performance, however, even the second version of the switch-on-residuals filter, which was partially developed by "cut and try" modifications to the original version, does not better the performance of the simpler, switch-on-range adaptive filter. The problem encountered with switch-on-residual filter seems to be that measurement noise causes the filter to switch to the "high Q" gain schedule when tracking long range constant-velocity targets.

The compromise filter, the simplest approach, provides fair performance -- better than the original version of the switch-on-residuals adaptive filter -- and serves as a standard for comparison with the adaptive filter performance.

B. SUGGESTIONS FOR FURTHER INVESTIGATION

One of the most obvious extensions of the simulation work is to incorporate the rotated coordinate system algorithm developed in Chapter II into the Monte-Carlo program for complete testing.

The switch-on-residuals adaptive filter should receive more attention. This is believed to be potentially the better of the adaptive filtering schemes but the most effective combination of gating levels and switching rules has not been found.

The switch-on-range adaptive filter should be simulated using other values for the steady state "high Q" gain matrix. A prime candidate is the steady-state gain matrix of the correlated-random-acceleration model with $\alpha=0.5$ and assumed forcing inputs of 0.1 g in the x-direction and one g in the y and z directions.

The various models should also be simulated using other types of trajectories including a proportional navigation trajectory representing an anti-ship missile attack.

More effort should be given to developing a definite quantitative correlation between prediction accuracy and estimation error statistics as discussed briefly in Chapter IV. A thorough investigation of relationships between the two sets of statistics may provide useful information for use in filter design.

C. CONCLUSIONS

The rotated cartesian coordinate system has been shown to have potential advantages over the fixed cartesian coordinate system both in diminished correlation of measurement noise and in selection of the components of assumed forcing inputs for the system model.

The prediction accuracy statistics are concise and valid measures of the system's performance in most cases, however the mean and covariance of estimation error are valuable statistics to have available to support the prediction accuracy statistics and to serve as measures of filter performance in cases such as the long range maneuvering target, where the prediction accuracy measurement provides little information.

The constant-velocity model with small values of assumed forcing inputs has been shown to provide the best performance against constant-velocity aircraft while the correlated-random-acceleration model with the proper choice of α to match the maneuver time constant provides the best performance against maneuvering targets.

Finally, the relatively simple switch-on-range adaptive filter and the more complex switch-on-residuals adaptive filter have been shown to provide improved performance over the compromise non-adaptive filter.

APPENDIX A

THE MONTE-CARLO SIMULATION PROGRAM

A.1 Program Description

The general purpose Monte-Carlo simulation program developed by Prof. D. E. Kirk of the Naval Postgraduate School is quite flexible in its basic form and offers several options to the user. The various options and many of the input data requirements are explained by comments early in the program listing.

A.2 The Monte-Carlo Simulation Program Listing

```
C     THIS PRCGRAM PERFCRMS MCNTE CARLC SIMULATICN OF STATE ESTIMATORS
C     CF WHICH THE KALMAN FILTER IS ONE EXAMPLE. THERE ARE SEVERAL
C     CPTICNS AVAILABLE AS INDICATED IN THE DETAI ED COMMENTS BELOW..
C     IT SHOULD BE NOTED THAT ALL COMPUTATIONS OF GAINS USING THE
C     SUBROUTINE GAIN ARE PERFORMED IN DOUBLE PRECISION. THUS ALL
C     ARRAYS FOR USE IN "GAIN" MUST BE PREPARED ACCCRDINGLY.
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
      DIMENSICN XP(160), YP(160), KHIT(160)
C
C
C     N=CRDER OF SYSTEM MODEL AND FILTER (DIMENSICN OF X,XHAT)
C
C     M=NUMBER OF MEASUREMENTS (DIMENSION OF THE VECTOR Z)
C
C     IN=NUMBER OF INPUT RANDOM FORCING FCNS (=DIMENSION OF W)
C
C     NSAM=NUMBER OF TIME SAMPLES
C
C     NENS=NUMBER OF MEMBERS IN ENSEMBLE
C
      READ (5,1) N,M,IN,NSAM,NENS
    1 FORMAT (5(I10))
C
C
C     THE VALUE OF ND READ IN MUST EQUAL THE ROW (AND COLUMN) DIMENSION
C     SPECIFIED FOR THE SQUARE MATRIX "TEMP1", E.G. IF TEMP1(3,3) IS
C     SPECIFIED IN THE COMMON STATEMENT "ND" MUST BE EQUAL TO 3.
      READ (5,2) ND
C
C     THE FLAG IADAPT CAUSES THE FILTER TO BE ADAPTIVE IF  IT IS NOT
C     ZERO. SUBROUTINE ADAPT MUST BE PROVIDED BY THE USER IF THIS OPTION
C     IS DESIRED.
C
      READ (5,2) IADAPT
    2 FORMAT (I2)
C
C
C     IG=-1 -- GAINS COMPUTED OFF-LINE AND READ IN
C         0 -- GAINS COMPUTED ONLY ONCE BEFORE STARTING MONTE CARLO
C         1 -- GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE
```

```
C
C     IFLR=0 -- R IS READ IN
C     IFLR.NE.0 -- R IS COMPUTED ON-LINE AT EACH TIME SAMPLE
C
C     IEST=0 -- STANDARD KALMAN FILTER EQUATION IS USED
C     IEST.NE.0 -- STD. KALMAN FILTER EQ. NOT USED
C
C     ITRK= 0 -- SEVERAL TRACKS GENERATED FROM STD LINEAR EQS
C     ITRK= 1 -- ONLY ONE TRACK IS USED
C      ITRK.NE.0OR1 -- SEVERAL TRACKS GENERATED BUT NOT FROM STD.
C                      LINEAR DIFFERENCE EQUATIONS
C
C     ISTAT=0 -- MEAN OF TRACK,MEAN & VARIANCE OF EST. ERROR COMPUTED
C     ISTAT.NE.0 -- SAME AS ISTAT=0 BUT OFF-DIAGONAL TERMS IN COVARIANCE
C
C                  MATRIX ARE ALSO COMPUTED
C
      READ (5,3) IG,IFLR,IEST,ITRK,ISTAT,IQ,ITRO,IMISS
    3 FORMAT (8(I10))
C
C     IG=-1 -- Q IS COMPUTED ON-LINE AT EACH SAMPLE BY "QON"
C         0 -- Q MATRIX IS READ IN
C         1 -- COVARIANCE OF W READ IN AND Q IS  COMPUTED ONCE BEFORE
C              MONTE CARLO BEGINS BY CALLING "QMAT"
C
C     ITRO=0 -- ONLY USED IF ONE TRACK; ITRO=0 INDICATES THE
C               TRACK IS READ IN (S.P.), OTHERWISE THE TRACK
C               IS GENERATED BY A SINGLE CALL OF SUBROUTINE
C               TRACK.
C
C
C     IMISS= 0---PROJECTILE MISS DISTANCE NOT COMPUTED. OTHERWISE IT IS
C                CALCULATED BY CALL OF SUBROUTINE BURST.
C
C     IPRT=0 -- SOME OR ALL OUTPUT DATA IS PRINTED
C
C     IPLT=0 -- SOME OR ALL OUTPUT DATA IS PLOTTED
      READ (5,4) IPRT,IPLT
    4 FORMAT (2(I5))
C
C     IGPLT,ITHVPL,IMTPLT,ISMPLT,ISVPLT ARE PLOTTING INDICATORS
C     IF THE INDICATOR IS EQUAL TO 1 PLOTS ARE GENERATED. THE INDICATORS
C     CORRESPOND TO THE FOLLOWING QUANTITIES
C     IGPLT(GAIN VS K),ITHVPL (THEORETICAL COVARIANCE MATRIX VS K),
C     IMTPLT (SAMPLE MEAN OF TRACK=TRACK IF ONLY ONE TRACK),
C     ISMPLT (SAMPLE MEAN OF EST. ERROR VS K), ISVPLT (SAMPLE VARIANCE
C     OF EST. ERROR VS K)
      IF (IPLT.EQ.0) READ (5,5) IGPLT,ITHVPL,IMTPLT,ISMPLT,ISVPLT
```

```
    5 FCRMAT (5(I10))
C
C     IPHI=0 -- PHI MATRIX TO BE READ IN (IN DCUBLE PREC)
C     IPHI.NE.0 -- PHI MATRIX NOT TO BE READ
C
C     IH=0 -- H MATRIX TO BE READ IN (IN D.P.)
C
C     IR=0 -- R MATRIX IS READ IN
C
C     IFKKM1=0 -- THE STARTING VALUE FCR THE MATRIX P(K/K-1) IS TO
C                 BE READ IN (D.P.)
C
C     IGAM=0 -- GAMMA MATRIX TO BE READ IN (D.P.)
C
C     ISIGV=0 -- STD. DEVIATICNS OF MEASUREMENT NGISE TO BE READ IN (SP)
C
C     ISIGW=0 -- STD. DEVIATIONS OF RANDCM INPUTS TO BE READ IN (SP)
C                SIGW MUST BE READ IN IF ITRK = 0.
C
C     IXHZ=0 -- INITIAL VALUE OF THE PREDICTED VALUE XH(0/-1) TO
C               BE READ IN (S.P.)
C
C     IC=1 -- ONE INITIAL CONDITION VALUE USED FOR THE STATE
C             OTHERWISE THE VALUE OF X(0) IS GENERATED BY A RANDOM
C             NIMBER GENERATOR CALLED BY "XZERC"
C
C
      READ (5,6) IPHI,IH,IR,IPKKM1,IGAM,ISIGV,ISIGW,IXHZ,IC
    6 FCRMAT (10I5)
      CALL CVFLOW
      IW = 6395217
      IV = 1936748
      IXZ = 135769
C
C
C     THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE RUN AS
C     SPECIFIED BY THE USER'S FLAGS
C
C
      WRITE (6,143)
      WRITE (6,7)
    7 FCRMAT (20X,'DESCRIPTION OF RUN',//)
      IF (IG.EQ.0) GO TO 9
      IF (IG.EQ.1) GO TO 11
      WRITE (6,8)
    8 FCRMAT (10X,'GAINS COMPUTED OFF-LINE AND READ IN',/)
      GC TO 13
    9 WRITE (6,10)
```

```
10 FORMAT (10X,'GAINS COMPUTED ONCE IN "GAIN" BEFORE STARTING MONTE C
  1ARLO',/)
   GO TO 13
11 WRITE (6,12)
12 FORMAT (10X,'GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE',/)
13 IF (IEST.EQ.0) GO TO 15
   WRITE (6,14)
14 FORMAT (10X,'THE STANDARD LINEAR EQS. DO NOT CHARACTERIZE THE FILT
  1ER',/)
   GO TO 17
15 WRITE (6,16)
16 FORMAT (10X,'THE STD. KALMAN EQS. CHARACTERIZE THE LINEAR FILTER',
  1/)
17 IF (ITRK.EQ.0) GO TO 23
   IF (ITRK.EQ.-1) GO TO 21
   IF (ITRO.EQ.0) GO TO 19
   WRITE (6,18)
18 FORMAT (10X,'ONLY ONE TRACK IS USED AND IT IS GENERATED BY SUBROUT
  1INE TRACK',/)
   GO TO 25
19 WRITE (6,20)
20 FORMAT (10X,'ONLY ONE TRACK IS USED AND IT IS READ IN',/)
   GO TO 25
21 WRITE (6,22)
22 FORMAT (10X,'SEVERAL TRACKS USED BUT NOT GENERATED FROM STD. LINEA
  1R DIFFERENCE EQS.',/)
   GO TO 25
23 WRITE (6,24)
24 FORMAT (10X,'SEVERAL TRACKS GENERATED BY USING THE STD. LINEAR DIF
  1FERENCE EQS.',/)
25 IF (ISTAT.EQ.0) GO TO 27
   WRITE (6,26)
26 FORMAT (10X,'MEAN OF TRACK, MEAN OF EST. ERROR AND COVARIANCE OF E
  1ST. ERROR ARE COMPUTED',/)
   GO TO 29
27 WRITE (6,28)
28 FORMAT (10X,'MEAN OF TRACK, MEAN AND VARIANCES OF EST. ERROR ARE
  1COMPUTED',/)
29 IF (IQ.EQ.0) GO TO 33
   IF (IQ.EQ.1) GO TO 31
   WRITE (6,30)
30 FORMAT (10X,'THE Q MATRIX IS COMPUTED ON-LINE AT EACH SAMPLE BY "Q
  1ON"',/)
   GO TO 35
31 WRITE (6,32)
32 FORMAT (10X,'THE COVARIANCE OF W IS READ IN AND Q IS COMPUTED BY "
  1QMAT" BEFORE STARTING MONTE CARLO',/)
   GO TO 35
```

```
   33 WRITE (6,34)
   34 FORMAT (10X,'THE Q MATRIX IS READ IN',/)
   35 IF (IFLR.EQ.0) GO TO 37
      WRITE (6,36)
   36 FORMAT (10X,'R IS COMPUTED ON-LINE AT EACH SAMPLE BY "RON"',/)
      GO TO 39
   37 WRITE (6,38)
   38 FORMAT (10X,'R IS READ IN',/)
   39 WRITE (6,40)
   40 FORMAT (////,20X,'INPUT DATA CALLED FOR',//)
      IF (IPHI.EQ.0) WRITE (6,41)
   41 FORMAT (10X,'PHI MATRIX',/)
      IF (IH.EQ.0) WRITE (6,42)
   42 FORMAT (10X,'H MATRIX',/)
      IF (IFLR.EQ.0) WRITE (6,43)
   43 FORMAT (10X,'R MATRIX',/)
      IF (IQ.EQ.0) GO TO 45
      IF (IQ.EQ.-1) GO TO 47
      WRITE (6,44)
   44 FORMAT (10X,'COVARIANCE OF W',/)
      GO TO 47
   45 WRITE (6,46)
   46 FORMAT (10X,'Q MATRIX',/)
   47 IF (IGAM.EQ.0) WRITE (6,48)
   48 FORMAT (10X,'GAMMA MATRIX',/)
      IF (ISIGV.EQ.0) WRITE (6,49)
   49 FORMAT (10X,'STANDARD DEVIATIONS OF MEASUREMENT NOISE',/)
      IF (ISIGW.EQ.0) WRITE (6,50)
   50 FORMAT (10X,'STANDARD DEVIATIONS OF INPUT FORCING W',/)
      IF (IXHZ.EQ.0) WRITE (6,51)
   51 FORMAT (10X,'XHAT(0/-1)',/)
      IF (IC.NE.1) WRITE (6,52)
   52 FORMAT (10X,'MEANS AND VARIANCES OF X(0)',/)
      IF (IPKKM1.EQ.0) WRITE (6,53)
   53 FORMAT (10X,'P(0/-1)',/)
C
C
C     THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE OUTPUT
C     DATA CALLED FOR
C
C
      WRITE (6,54)
   54 FORMAT (////,20X,'OUTPUT CALLED FOR',//)
      IF (IPRT.NE.0) GO TO 60
      WRITE (6,55)
   55 FORMAT (10X,'PRINTED OUTPUT OF THE FOLLOWING DATA',/)
      IF (IG.EQ.0) WRITE (6,56)
   56 FORMAT (15X,'GAIN MATRICES AND THEORETICAL COVARIANCE OF EST. ERRO
```

```
   1R MATRICES',/)                                                        2410
    WRITE (6,57)                                                          2420
 57 FORMAT (15X,'SAMPLE MEANS OF TRACK AND ESTIMATION ERROR,SAMPLE VAR    2430
   1IANCES CF EST. ERROR',/)                                              2440
    IF (ISTAT.NE.0) WRITE (6,58)                                          2450
 58 FCRMAT (15X,'COVARIANCE OF ESTIMATION ERROR MATRIX',/)                2460
    IF (IMISS.NE.0) WRITE (6,59)                                          2470
 59 FCRMAT (15X,'MEAN PROJECTILE MISS DISTANCE FOR EACH SAMPLE ',/)       2480
    GC TO 62                                                              2490
 6C WRITE (6,61)                                                          2500
 61 FCRMAT (10X,'NO PRINTED OUTPUT CALLED FOR',//)                        2510
 62 IF (IPLT.NE.0) GO TO 69                                               2520
    WRITE (6,63)                                                          2530
 63 FCRMAT (10X,'THE FOLLCWING PLOTS ARE CALLED FOR',/)                   2540
    IF (IGPLT.EQ.1) WRITE (6,64)                                          2550
 64 FCRMAT (15X,'G(K) VS. K',/)                                           2560
    IF (ITHVPL.EQ.1) WRITE (6,65)                                         2570
 65 FCRMAT (15X,'P(K/K) THEORETICAL VS. K',/)                             2580
    IF (IMTPLT.EQ.1) WRITE (6,66)                                         2590
 66 FCRMAT (15X,'MEAN OF TRACK VS. K',/)                                  2600
    IF (ISMPLT.EQ.1) WRITE (6,67)                                         2610
 67 FORMAT (15X,'SAMPLE MEANS OF ESTIMATION ERRCR VS. K',/)               2620
    IF (ISVPLT.EQ.1) WRITE (6,68)                                         2630
 68 FORMAT (15X,'SAMPLE VARIANCES OF ESTIMATION ERROR VS. K',/)           2640
    GC TO 71                                                              2650
 69 WRITE (6,70)                                                          2660
 70 FCRMAT (10X,'NO PLOTS CALLED FOR',/)                                  2670
 71 CCNTINUE                                                              2680
    WRITE (6,143)                                                         2690
    WRITE (6,72)                                                          2700
 72 FCRMAT (20X,'INPUT DATA',///)                                         2710
    WRITE (6,73) N,M,IN,NSAM,NENS                                         2720
 73 FCRMAT (4X,'N=',I2,4X,'M=',I2,4X'IN=',I2,4X'NSAM=',I3,4X,'NENS=',I    2730
   15,//)                                                                 2740
C                                                                         2750
C                                                                         2760
C   THE FOLLOWING SECTION READS THE SPECIFIED INPUT MATRICES              2770
C                                                                         2780
    IF (IPHI.NE.0) GO TO 76                                               2790
    CALL MREAD (PHI,N,N)                                                  2800
C                                                                         2810
    DC 74 I=1,N                                                           2820
C                                                                         2830
    DC 74 J=1,N                                                           2840
 74 PFIS(I,J) = PHI(I,J)                                                  2850
C                                                                         2860
    WRITE (6,75)                                                          2870
 75 FCRMAT (//,10X,'THE PHI MATRIX IS',/)                                 2880
```

```
      CALL MWRITE (PHI,N,N)
C
C
   76 IF (IH.NE.0) GO TO 79
      CALL MREAD (H,M,N)
C
      DO 77 I=1,M
C
      DC 77 J=1,N
   77 HS(I,J) = H(I,J)
C
      WRITE (6,78)
   78 FCRMAT (//,10X,'THE H MATRIX IS',/)
      CALL MWRITE (H,M,N)
C
C
   79 IF (IR.NE.0) GO TO 81
      CALL MREAD (R,M,M)
      WRITE (6,80)
   80 FORMAT (//,10X,'THE R MATRIX IS',/)
      CALL MWRITE (R,M,M)
C
C
   81 IF (IQ.NE.1) GO TO 83
      CALL MREAD (COVW,IN,IN)
      WRITE (6,82)
   82 FORMAT (//,10X,'THE COVARIANCE CF W MATRIX IS',/)
      CALL MWRITE (COVW,IN,IN)
      GC TO 35
   83 IF (IQ.NE.0) GO TO 85
      CALL MREAD (Q,N,N)
      WRITE (6,84)
   84 FCRMAT (//,10X,'THE Q MATRIX IS',/)
      CALL MWRITE (Q,N,N)
C
C
   85 IF (IGAM.NE.0) GO TO 88
      CALL MREAD (GAMMA,N,IN)
C
      DC 86 I=1,N
C
      DC 86 J=1,IN
   86 GAMMAS(I,J) = GAMMA(I,J)
C
      WRITE (6,87)
   87 FCRMAT (//,10X,'THE GAMMA MATRIX IS',/)
      CALL MWRITE (GAMMA,N,IN)
C
```

```
C
   88 IF (IPKKM1.NE.0) GO TO 90
      CALL MREAD (PKKM1,N,N)
      WRITE (6,89)
   89 FCRMAT (//,10X,'THE MATRIX P(0/-1) IS',/)
      CALL MWRITE (PKKM1,N,N)
C
C
   90 IF (ISIGV.NE.0) GO TO 92
      CALL VREAD (SIGV,M)
      WRITE (6,91)
   91 FCRMAT (//,10X,'THE STD. DEVIATICNS OF MEASLREMENT NCISE ARE',/)
      CALL VWRITE (SIGV,M)
C
C
   92 IF (ISIGW.NE.0) GO TO 94
      CALL VREAD (SIGW,IN)
      WRITE (6,93)
   93 FORMAT (//,10X,'THE STD. DEVIATICNS OF INPUT FORCING W ARE',/)
      CALL VWRITE (SIGW,IN)
C
C
   94 IF (IXHZ.NE.0) GO TO 96
      CALL VREAD (XHATZ,N)
      WRITE (6,95)
   95 FCRMAT (//,10X,'THE VECTOR XHAT(0/-1) IS',/)
      CALL VWRITE (XHATZ,N)
C
C
   96 IF (IC.EQ.1) GO TO 99
C     IC.NE.1 MEANS THAT MEANS AND STD. DEVIATIONS OF THE INITIAL STATE
C     VALUE MUST BE READ IN. OTHERWISE NOT READ IN.
      CALL VREAD (XZMEAN,N)
      WRITE (6,97)
   97 FCRMAT (//,10X,'THE MEAN OF THE VECTOR X(0) IS',/)
      CALL VWRITE (XZMEAN,N)
C
C
      CALL VREAD (SIGXZ,N)
      WRITE (6,98)
   98 FOPMAT (//,10X,'THE STANDARD DEVIATIONS OF THE VECTOR X(0) ARE',/)
      CALL VWRITE (SIGXZ,N)
      GC TO 107
C
C
   99 READ (5,102) (XS(I,1),I=1,N)
C     INITIAL CONDITION HAS BEEN READ
      WRITE (6,100)
```

```
  100 FORMAT (//,10X,'THE INITIAL STATE IS',/)
      WRITE (6,106) (XS(I,1),I=1,N)
      IF (ITRK.NE.1) GO TO 107
      IF (ITRC.NE.0) GO TO 103
C
      DO 101 K=2,NSAM
  101 READ (5,102) (XS(I,K),I=1,N)
C
  102 FORMAT (4F20.12)
      GO TO 104
  103 CALL TRACK
C     IF TRACK CALLED HERE IT SHOULD BE WRITTEN TO GENERATE AND
C     STORE THE TRACK IN XS(N,K) FOR K=2(NOTE),NSAM
C
  104 WRITE (6,105)
  105 FORMAT (//,10X,'THE FIRST AND LAST POINTS ON THE SINGLE TRACK TO B
     1E USED ARE',/)
      WRITE (6,106) (XS(I,1),I=1,N)
      WRITE (6,106) (XS(I,NSAM),I=1,N)
  106 FORMAT (9(2X,1PE12.5),/)
  107 CONTINUE
C
C
C     THE FOLLOWING SECTION PREPARES FOR THE MONTE CARLO LOOP
C
C     FORM NXN IDENTITY MATRIX IN DOUBLE PRECISION
C
      DO 108 I=1,N
C
      DO 108 J=1,N
      EI(I,J) = 0.D0
  108 IF (I.EQ.J) EI(I,J)=1.D0
C
C     GIVEN THE MATRIX GAMMA AND THE COVARIANCE OF W COMPUTE Q
C     USING DOUBLE PRECISION ARITHMETIC
      IF (IQ.NE.1) GO TO 109
      CALL QMAT
      WRITE (6,84)
      CALL MWRITE (Q,N,N)
  109 IF (IG.NE.-1) GO TO 112
C
      DO 110 K=1,40
C
      DO 110 I=1,N
  110 READ (5,102) (GKS(I,J,K),J=1,M)
C
C
      DO 111 K=41,NSAM
```

```
C
      DO 111 I=1,N
C
      DO 111 J=1,M
  111 GKS(I,J,K) = GKS(I,J,40)
C
      GO TO 115
  112 IF (IG.NE.0) GO TO 115
C
      DO 114 K=1,NSAM
      CALL GAIN
C
      DO 114 I=1,N
C
      DO 113 L=1,N
  113 PKKS(I,L,K) = PKK(I,L)
C
C
      DO 114 J=1,M
  114 GKS(I,J,K) = G(I,J)
C
  115 CONTINUE
C     IF GAINS WERE TO BE READ IN (IG=-1) OR COMPUTED ONLY
C     ONCE(IG=0), THIS HAS NOW BEEN DONE
C
C
C     SET UP ARRAYS FOR COMPUTING STATISTICS
      NFIRE = 0
      NHIT = 0
      NSCARE = 0
C
      DO 116 K=1,NSAM
      AMISS(K) = 0.0
      KHIT(K) = 0
C
      DO 116 J=1,N
      XM(J,K) = 0.
      ERR(J,K) = 0.
C
      DO 116 L=1,N
  116 VAR(J,L,K) = 0.
C
C
C
C     BEGIN MAIN ITERATION LOOP HERE
C
      DO 123 ITER=1,NENS
      IF (IC.EQ.1) GO TO 117
```

```
      CALL XZERO
C
  117 DO 118 I=1,N
  118 XHKKM1(I) = XHATZ(I)
C
C
      DO 123 K=1,NSAM
      FORM NOISY MEASUREMENT FROM TRUE STATE VALUE
C
C
C
      DO 119 I=1,N
  119 X(I) = XS(I,K)
C
      CALL MEAS
C     GAIN IS NOT TO BE COMPUTED ON-LINE IF IG.NE.1
      IF (IG.NE.1) GO TO 121
C
C     Q IS TO BE COMPUTED ON-LINE IF IFLQ.NE.0
      IF (IFLQ.NE.0) CALL QON
C     R IS TO BE COMPUTED ON-LINE IF IFLR.NE.0
C
      IF (IFLR.NE.0) CALL RON
      CALL GAIN
C
      DO 120 I=1,N
C
      DO 120 J=1,M
  120 GKS(I,J,K) = G(I,J)
C
C
C     UPDATE THE STATE ESTIMATE
C
  121 CALL ESTIM
C
C     UPDATE RUNNING SUMS USED IN COMPUTING STATISTICS
      CALL STAT
      IF (IMISS.EQ.0) GO TO 122
      CALL BURST (KHIT)
  122 IF (K.EQ.NSAM) GO TO 123
C
C     UPDATE TRACK BY COMPUTING X(K+1)
      IF (ITRK.NE.1) CALL TRACK
  123 CONTINUE
C
C
C     DIVIDE RUNNING SUMS COMPUTED BY SUBROUTINE STAT BY ENSEMBLE
C     SIZE TO COMPUTE STATISTICS
      ENS = NENS
```

```
C
      DO 125 K=1,NSAM
C
      DO 125 J=1,N
      IF (ITRK.EQ.1) GO TO 124
      XM(J,K) = XM(J,K)/ENS
  124 ERR(J,K) = ERR(J,K)/ENS
  125 VAR(J,J,K) = VAR(J,J,K)/ENS-ERR(J,K)**2
C
      IF (ISTAT.EQ.0) GO TO 127
C
C     COMPUTE OFF-DIAGONAL TERMS IN COVARIANCE OF ESTIMATION
C     ERROR MATRIX IF ISTAT.NE.0
C
      DO 126 K=1,NSAM
C
      DO 126 L=2,N
      LM1 = L-1
C
      DO 126 J=1,LM1
  126 VAR(L,J,K) = VAR(L,J,K)/ENS-ERR(L,K)*ERR(J,K)
C
  127 CONTINUE
C
      IF (IMISS.EQ.0) GO TO 129
C     DIVIDE RUNNING SUM OF MISS DISTANCES COMPUTED BY SUBROUTINE
C     BURST BY ENSEMBLE SIZE TO COMPUTE AVERAGE MISS DISTANCES.
C
      DO 128 K=1,NSAM
      AMISS(K) = AMISS(K)/ENS
      FIRE = FLOAT(NFIRE)/ENS
      HIT = FLOAT(NHIT)/ENS
  128 SCARE = FLOAT(NSCARE)/ENS
C
C
  129 IF (IPRT.NE.0) GO TO 151
      WRITE (6,130)
  130 FORMAT ('1',20X,'OUTPUT DATA',///)
C     WRITE GAINS, THEORETICAL COVARIANCES OF ESTIMATION ERROR
C     IF ONE SET OF GAINS HAS BEEN USED.
      IF (IADAPT.NE.0) GO TO 137
      IF (IG.NE.0) GO TO 137
      WRITE (6,131)
  131 FORMAT (10X,'THE GAIN MATRICES ARE',/)
C
      DO 133 K=1,NSAM
      WRITE (6,132) K
  132 FORMAT (5X,'K=',I3,/,10X,'G(K)=',/)
```

```
C
      DO 133 I=1,N
  133 WRITE (6,106) (GKS(I,J,K),J=1,M)
C
      WRITE (6,134)
  134 FORMAT (1X,//,10X,'THE THEORETICAL COVARIANCE MATRIX IS',/)
C
      DO 136 K=1,NSAM
      WRITE (6,135) K
  135 FORMAT (5X,'K=',I3,/,10X,'P(K/K)=',/)
C
      DO 136 I=1,N
  136 WRITE (6,106) (PKKS(I,J,K),J=1,N)
C
C
C
  137 WRITE (6,143)
      WRITE (6,138)
      WRITE (6,139)
  138 FORMAT (T5,'TIME',T16,'VECTOR COM-',T34,'SAMPLE MEAN',
     1 T51,'SAMPLE MEAN OF',T71,'SAMPLE VARIANCE OF')
  139 FORMAT (T5,'INDEX',T16,'PONENT INDEX',T34,'OF TRACK',
     1 T51,'ESTIMATION ERROR',T71,'ESTIMATION ERROR')
  140 FORMAT (6X,I3,13X,I1,10X,1PE14.7,2(6X,1PE14.7))
  141 FORMAT (//)
C
      DO 142 K=1,NSAM
      WRITE (6,141)
C
      DO 142 I=1,N
  142 WRITE (6,140) K,I,XM(I,K),ERR(I,K),VAR(I,I,K)
C
      WRITE (6,143)
      IF (ISTAT.EQ.0) GO TO 147
  143 FORMAT ('1')
      WRITE (6,144)
  144 FORMAT (10X,'THE SAMPLE COVARIANCE OF EST. ERROR MATRIX IS',//)
C
      DO 145 K=1,NSAM
      WRITE (6,146) K
C
      DO 145 I=1,N
  145 WRITE (6,106) (VAR(I,L,K),L=1,I)
C
  146 FORMAT (//,2X,'K=',I3,/)
      WRITE (6,143)
  147 IF (IMISS.EQ.0) GO TO 151
C
```

```
      DO 148 K=1,NSAM
  148 WRITE (6,149) K,AMISS(K),K,KHIT(K)
C
  149 FORMAT (' ',T10,'AMISS(',I3,') = ',G20.12,10X,'KHIT(',I3,') = ',I4
     1)
      WRITE (6,150) FIRE,HIT,SCARE
  150 FORMAT ('0',T10,'AVE. NO. FIRED = ',G12.4,5X,'AVE. NO. HITS = ',G1
     12.4,5X,'AVE. NO. SCARES AND/OR HITS = ',G12.4)
      WRITE (6,143)
  151 IF (IPLT.NE.0) GO TO 172
C
      DO 152 K=1,NSAM
  152 XP(K) = K
C
      IF (IGPLT.NE.1) GO TO 156
C
      DO 154 I=1,N
C
      DO 154 J=1,M
C
      DO 153 K=1,NSAM
  153 YP(K) = GKS(I,J,K)
C
      WRITE (6,143)
      CALL PLOTP (XP,YP,NSAM,0)
  154 WRITE (6,155) I,J
C
  155 FORMAT (12X,'G(',I1,',',I1,')  VS. K')
  156 IF (ITHVPL.NE.1) GO TO 160
C
      DO 158 I=1,N
C
      DO 157 K=1,NSAM
  157 YP(K) = PKKS(I,I,K)
C
      WRITE (6,143)
      CALL PLOTP (XP,YP,NSAM,0)
  158 WRITE (6,159) I,I
C
  159 FORMAT (12X,'PKK(',I1,',',I1,')  VS. K')
  160 IF (IMTPLT.NE.1) GO TO 164
C
      DO 162 I=1,N
C
      DO 161 K=1,NSAM
  161 YP(K) = XM(I,K)
C
      WRITE (6,143)
```

```
      CALL PLOTP (XP,YP,NSAM,0)
  162 WRITE (6,163) I
C
  163 FCRMAT (12X,'MEAN OF X(',I1,')  VS. K')
  164 IF (ISMPLT.NE.1) GO TO 168
C
      DC 166 I=1,N
C
      DO 165 K=1,NSAM
  165 YP(K) = ERR(I,K)
C
      WRITE (6,143)
      CALL PLCTP (XP,YP,NSAM,0)
  166 WRITE (6,167) I,I
C
  167 FCRMAT (12X,'XHATKK(',I1,') -X(',I1,')  VS. K')
  168 IF (ISVPLT.NE.1) GO TO 172
C
      DC 170 I=1,N
C
      DC 169 K=1,NSAM
  169 YP(K) = VAR(I,I,K)
C
      WRITE (6,143)
      CALL PLOTP (XP,YP,NSAM,0)
  170 WRITE (6,171) I
C
  171 FORMAT (12X,'ERROR VARIANCE(',I1,') VS.. K')
      WRITE (6,143)
  172 CCNTINUE
      STCP
      END
C
C
C
      SUBROUTINE ADAPT
C
C        THIS SUBROUTINE CCNTROLS THE ACAPTIVE SWITCHING OF THE FILTER
C        PARAMETERS IN ORDER TO MAKE THE FILTER MCRE RESPONSIVE IN A
C        CHANGING SITUATION.
C
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
```

```
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
      DIMENSION GKA(9,9,41)
C
C     THE USER MUST SUPLY STATEMENTS HERE TO CONTROL THE ADAPTIVE
C     FILTER OPERATION.
C
C
   18 RETURN
      END
C
C
C
      SUBROUTINE BURST (KHIT)
C
C        THIS SUBROUTINE COMPUTES PROJECTILE TIME OF FLIGHT AND OUTPUTS
C        A SET OF  STATISTICS INDICATING THE ACCURACY OF THE PREDICTED
C        STATE VALUES WHEN THE TARGET IS WITHIN GUN RANGE. AS WRITTEN
C        THIS SUBROUTINE IS VALID FOR CARTESIAN COORDINATES ONLY.
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
      DIMENSION XEST(9,160), XF(9), TEMPP(9,9), KHIT(160)
      RING1 = 8.0
      RING2 = 16.0
      SUMSQ = 0.0
      ERRSQ = 0.0
      TSAMP = 0.25
      TOFCON = 1.4285E-03
C
C     TOFCON REPRESENTS THE SLOPE OF A STRAIGHT LINE APPROXIMATION OF
C     THE TIME OF FLIGHT (TOF) VS RANGE CURVE FOR A 5 IN./54 CAL. GUN
C     (SEC/YARD)
C
      N1 = N/3
      N2 = 2*N1
      NN = N-1
C
      DO 1 I=1,N
C
C     STORE THE CURRENT ESTIMATE FOR LATER USE IN CALCULATING THE
C     PREDICTION.
```

```
C
      1 XEST(I,K) = XHKK(I)
C
C
        DC 2 I=1,NN,N1
C
C       CALCULATE THE RANGE ESTIMATE FROM CARTESIAN CORDINATE POSITION
C       STATE ESTIMATES.
C
      2 SUMSQ = SUMSQ+XHKK(I)**2
C
        D = SQRT(SUMSQ)
C
C       TIME OF FLIGHT IS DIRECTLY PROPORTIONAL TO RANGE
C
        TCF = D*TOFCON
C
C       CALCULATE THE NUMBER OF SAMPLES WHICH WILL BE TAKEN DURING THE
C       PROJECTILE TIME OF FLIGHT,AND STORE AS VARIABLE INTOF.
C
        INTOF = TOF/TSAMP
        IF (INTOF.GE.K) GO TO 3
C
C       IF T. C. F. IS GREATER THAN 10 SEC. THE TARGET IS BEYOND THE
C       MAXIMUM EFFECTIVE FIRING RANGE .
C
        IF ((INTOF*TSAMP).LE.10.0) GO TO 4
C
C       IF THE TARGET IS NOT FIRED UPON, SET IN A LARGE VALUE OF MISS
C       DISTANCE AS AN INDICATOR.
C
      3 AMISS(K) = AMISS(K)+9999.9
        RETURN
C
C       CCUNT THE NUMBER OF SHOTS FIRED PER RUN.
C
      4 NFIRE = NFIRE+1
C
        DC 5 I=1,N
C
C       THE PREDICTION FOR THE GUNFIRE SCLUTION WAS MADE FROM AN ESTIMATE
C       WHICH WAS GENERATED TOF SECONDS EARLIER.
C
      5 XF(I) = XEST(I,K-INTOF)
C
C
C       THE FOLLOWING ALGORITHM (THRU STATEMENT NO. 23) CALCULATES THE
C       STATE PREDICTION TRANSITION MATRIX, PHI**INTOF.
```

```
C
      SUM1 = 0.0
      SUM2 = 0.0
      TOF = FLOAT(INTOF)
C
      DO 6 I=1,N
C
      DO 6 J=1,N
    6 TEMPP(I,J) = 0.0
C
      TEMPP(1,1) = 1.0
      TEMPP(1,2) = TSAMP*TOF
      TEMPP(2,2) = 1.0
      IF (N.NE.9) GO TO 9
C
      DO 7 I=1,INTOF
    7 SUM1 = SUM1+PHIS(3,3)**(I-1)
C
      JTOF = INTOF-1
C
      DO 8 I=1,JTOF
    8 SUM2 = SUM2+(INTOF-I)*PHIS(3,3)**(I-1)
C
      TEMPP(1,3) = PHIS(1,3)*SUM1+TSAMP*PHIS(2,3)*SUM2
      TEMPP(2,3) = PHIS(2,3)*SUM1
      TEMPP(3,3) = PHIS(3,3)**INTOF
    9 IF (N.LT.6) GO TO 11
C
      DO 10 NA=N1,N2,N1
C
      DO 10 I=1,N1
      INA = I+NA
C
      DO 10 J=1,N1
      JNA = J+NA
   10 TEMPP(INA,JNA) = TEMPP(I,J)
C
   11 CONTINUE
C
C     XHAT(K/K-INTOF)=(PHI**INTOF)*XHAT(K-INTOF/K-INTOF)
C
      CALL VPROD (TEMPP,XF,N,N,VTMP)
C
      DO 12 I=1,NN,N1
C
C     PREDICTION ERROR = X(K)-XHAT(K/K-INTOF)
C
      ERD = XS(I,K)-VTMP(I)
```

```
   12 ERRSQ = ERRSQ+ERD**2
C
C
C     MISS DISTANCE = SQRT(SUM OF SQUARED POSITION ERRORS)
C
      BURST = SQRT(ERRSQ)
      AMISS(K) = AMISS(K)+BURST
C
C     IF THE MISS DISTANCE IS LESS THAN THE RADIUS RING2 THE PREDICTION
C     IS CONSIDERED TO BE A NEAR MISS OR SCARE.
C
      IF (BURST.GT.RING2) GO TO 13
C
C     COUNT THE NUMBER OF HITS AND THE NUMBER OF SCARES PER RUN.
C
      NSCARE = NSCARE+1
      KHIT(K) = KHIT(K)+1
C
C     COUNT THE NUMBER OF HITS AND/OR SCARES PER SAMPLE OVER THE
C     ENSEMBLE.
C
C     IF THE MISS DISTANCE IS LESS THAN THE RADIUS RING1 THE PREDICTION
C     IS CONSIDERED TO BE A HIT.
C
      IF (BURST.LE.RING1) NHIT=NHIT+1
   13 RETURN
      END
C
C
C
      SUBROUTINE ESTIM
C
C     THIS SUBROUTINE UPDATES THE STATE ESTIMATE. IN THE DEFAULT
C     CONDITION (IEST.EQ.0) THE STANDARD EQUATIONS
C
C     XHAT(K/K)=XHAT(K/K-1)+G(K)*(Z(K)-H(K)*XHAT(K/K-1))
C
C     XHAT(K+1/K)=PHI*XHAT(K/K)
C
C     ARE EVALUATED
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
```

```
C     TAKE THE APPROPRIATE GAIN AND STORE IN THE ARRAY GK
C
      DO 1 I=1,N
C
      DO 1 J=1,M
    1 GK(I,J) = GKS(I,J,K)
C
      IF (IADAPT.NE.0) CALL ADAPT
      IF (IEST.NE.0) GO TO 2
      CALL VPROD (HS,XHKKM1,M,N,VTMP)
      CALL VSUB (Z,VTMP,M,VTMP)
      CALL VPROD (GK,VTMP,N,M,VTMP)
      CALL VADD (XHKKM1,VTMP,N,XHKK)
C
C     XHAT(K/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKK
      CALL VPROD (PHIS,XHKK,N,N,XHKKM1)
C
C     XHAT(K+1/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKKM1
      RETURN
    2 CONTINUE
C     IF STANDARD EQUATIONS ARE NOT TO BE USED, THE APPROPRIATE
C     EQUATIONS MUST BE INSERTED HERE BY THE USER.
      RETURN
      END
C
C
C
      SUBROUTINE GAIN
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
C
C     G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)
      CALL TRANS (H,M,N,TEMP2)
      CALL PROD (PKKM1,TEMP2,N,N,M,TEMP)
      CALL PROD (H,TEMP,M,N,M,TEMP1)
      CALL ADD (TEMP1,R,M,M,TEMP1)
      IF (M.EQ.1) GO TO 2
      MD = ND
      CALL GAUSS3 (M,EPS,TEMP1,TEMP2,KER,MD)
      CALL PROD (TEMP,TEMP2,N,M,M,G)
C
```

```
C      NCTE HERE   PKK(I,J) = P(K/K)   WHERE
C      P(K/K) = (I-G(K)*H)*P(K/K-1)
     1 CALL PROD (G,H,N,M,N,TEMP)
       CALL SUB (EI,TEMP,N,N,TEMP2)
       CALL PROD (TEMP2,PKKM1,N,N,N,PKK)
C
C      NCTE HERE   PKKM1(I,J) = P(K/K-1)   WHERE
C      P(K/K-1)= PHI*P(K-1/K-1)*PHIT + Q
       CALL TRANS (PHI,N,N,TEMP2)
       CALL PRCD (PKK,TEMP2,N,N,N,TEMP)
       CALL PROD (PHI,TEMP,N,N,N,TEMP1)
       CALL ADD (TEMP1,Q,N,N,PKKM1)
       RETURN
C
     2 DC 3 I=1,N
     3 G(I,1) = TEMP(I,1)/TEMP1(1,1)
C
       GC TO 1
       END
C
C
C
       SUBROUTINE MEAS
C
C      THIS SUBROUTINE STARTS WITH THE TRUE STATE VALUE XS
C      AND ADDS ZERO-MEAN WHITE GAUSSIAN NOISE TO H*XS TO
C      GENERATE A NOISY VERCTOR OF MEASUREMENTS Z.
C
C
       REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
       CCMMCN EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
      19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
      2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
      3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
      4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
      5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
      6,NSCARE,ICOORD,IADAPT
C
C          THE FOLLCWING ALGORITHM IS VALID FOR 1,2, OR 3 DIMENSIONAL
C          SYSTEMS WHEN MEASUREMENTS ARE TAKEN OF PCSITION CNLY. FOR CTHER
C          CASES IT MUST BE REVISED.
C
       CALL VPROD (HS,X,M,N,Z)
       IF (M-2) 1,2,3
     1 Z(2) = C.0
     2 Z(3) = 0.0
     3 Z1 = SQRT(Z(1)**2+Z(2)**2+Z(3)**2)
       IF (M-2) 6,5,4
```

```
    4 Z(3) = ATAN2(Z(3),Z(1))
    5 Z(2) = ATAN2(Z(2),Z(1))
    6 Z(1) = Z1
      CALL SNORM (IV,V,M)
C
      DO 7 I=1,M
    7 V(I) = SIGV(I)*V(I)
C
      CALL VADD (Z,V,M,Z)
      ALFA = Z(1)*COS(Z(2))*COS(Z(3))
      BETA = Z(1)*SIN(Z(2))*COS(Z(3))
      GAMMO = Z(1)*SIN(Z(3))
      Z(1) = ALFA
      Z(2) = BETA
      Z(3) = GAMMO
      END
C
C
C
      SUBROUTINE QMAT
C
C     THIS SUBROUTINE COMPUTES THE MATRIX Q FROM THE EQUATION
C
C        Q=GAMMA* E(W*WT) * GAMMAT
C
C     DOUBLE PRECISION ARITHMETIC IS USED
C
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
C
C
      CALL PROD (GAMMA,COVW,N,IN,IN,TEMP)
      CALL TRANS (GAMMA,N,IN,TEMP1)
      CALL PROD (TEMP,TEMP1,N,IN,N,Q)
      RETURN
      END
C
C
C
      SUBROUTINE QON
C
```

```
C
C     IF Q IS TO BE COMPUTED ON-LINE (IFLQ.NE.0) IT IS DONE
C     IN THIS SUBROUTINE
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
C
C     THE APPROPRIATE STATEMENTS FOR COMPUTING Q ON-LINE MUST
C     BE INSERTED HERE BY THE USER
C
      RETURN
      END
C
C
C
      SUBROUTINE RON
C
C     IF R IS TO BE COMPUTED ON-LINE (IFLR.NE.0) IT IS DONE
C     IN THIS SUBROUTINE
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
C
C     THE APPROPRIATE STATEMENTS FOR COMPUTING R ON-LINE MUST
C     BE INSERTED HERE BY THE USER
C
      RETURN
      END
C
C
C
      SUBROUTINE STAT
C
C     THIS SUBROUTINE  COMPUTES RUNNING SUMS USED IN DETERMINING THE
C     SAMPLE STATISTICS OF TRACK AND ESTIMATION ERRORS. IN THE DEFAULT
C     OPTION (ISTAT.EQ.0) THE STATISTICS TO BE COMPUTED ARE MEAN OF
C     TRACK, MEAN OF ESTIMATION ERROR AND VARIANCE OF ESTIMATION
```

```
C     ERROR. IF(ISTAT.NE.O) THE OFF-DIAGONAL TERMS IN THE COVARIANCE OF
C     ESTIMATION ERROR MATRIX ARE ALSO COMPUTED.
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      CCMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
      DIMENSION EXH(9)
      IF (ITRK.NE.1) GO TO 2
      IF (ITER.NE.1) GO TO 4
C
      DC 1 J=1,N
    1 XM(J,K) = XS(J,K)
C
      GC TO 4
    2 CCNTINUE
C
      DC 3 J=1,N
    3 XM(J,K) = XM(J,K)+XS(J,K)
C
    4 CCNTINUE
C
      DC 5 J=1,N
      EXH(J) = XHKK(J)-XS(J,K)
      ERR(J,K) = ERR(J,K)+EXH(J)
    5 VAR(J,J,K) = VAR(J,J,K)+EXH(J)**2
C
      IF (ISTAT.EQ.0) RETURN
C
      DC 6 L=2,N
      LM1 = L-1
C
      DC 6 J=1,LM1
    6 VAR(L,J,K) = VAR(L,J,K)+EXH(L)*EXH(J)
C
      RETURN
      END
C
C
C
      SUBRCUTINE TRACK
C
C     IF TRACK IS TO BE GENERATED ON-LINE IT IS DCNE IN THIS SLBROUTINE
C     IN THE DEFAULT OPTION (ITRK.EQ.0) THE TRACK IS GENERATED
```

```
C     FROM THE STANDARD LINEAR DIFFERENCE EQUATION
C
C     X(K+1)=PHI*X(K)+GAMMA*W(K)
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      CCMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
      DIMENSION W(3)
C     ITRK NE.0 CR 1 -- SEVERAL TRACKS GENERATED,BUT NOT FROM STD.
C                       LINEAR EQS.
C          = 0 -- SEVERAL TRACKS GENERATED FROM STC LINEAR EQS
C          = 1 -- ONLY ONE TRACK IS USED
      IF (ITRK.NE.0) GO TO 4
      CALL SNORM (IW,W,IN)
C     CCNVERT EACH N(0,1) R.V. TO N(0,SIGW(I)) R.V.
C
      DC 1 I=1,IN
    1 W(I) = SIGW(I)*W(I)
C
C
      DC 2 I=1,N
    2 X(I) = XS(I,K)
C
      CALL VPROD (GAMMAS,W,N,IN,W)
      CALL VPROD (PHIS,X,N,N,VTMP)
      CALL VADD (VTMP,W,N,VTMP)
C
      DC 3 I=1,N
    3 XS(I,K+1) = VTMP(I)
C
C     NEW VALUE OF X HAS BEEN COMPUTED AND STORED IN THE ARRAY XS
      RETURN
    4 IF (ITRK.NE.1) GO TO 6
C      IF(ITRK.EQ.1) THE USER MUST INSERT HERE THE STATEMENTS REQUIRED
C     TO GENERATE A SINGLE TRAJECTORY AND STORE IT IN THE ARRAY
C     XS(I,K),I=1,N,K=2,NSAM (NOTE THAT IF A SINGLE TRAJECTORY IS TO BE
C     GENERATED, THE INITIAL CONDITION HAS BEEN READ IN AND STORED
C     IN XS(I,1),I=1,N)
C
      DC 5 K=2,NSAM
      EKM1 = K-1
      T = 0.5*EKM1
      XS(1,K) = 13.-0.0173205*T
```

```
      XS(2,K) = -0.0173205
      XS(3,K) = 7.7-0.01*T
    5 XS(4,K) = -0.01
C
      RETURN
    6 CONTINUE
C     IF THIS POINT IS REACHED, ITRK NCT EQUAL 0 CR 1 INDICATING THAT
C     SEVERAL TRACKS ARE TO BE GENERATED, BUT NOT BY USING THE STD.
C     LINEAR DIFFERENCE EQS..THE USER MUST SUPPLY THE APPROPRIATE
C     STATEMENTS HERE.
      RETURN
      END
C
C
C
      SUBROUTINE XZERO
C
C     THIS SUBROUTINE GENERATES THE INITIAL STATE VALUE FROM A NORMAL
C     RANDOM NUMBER GENERATOR. IT IS ASSUMED THAT THE INITIAL STATE
C     HAS COMPONENTS THAT ARE INDEPENDENT
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
      CALL SNORM (IXZ,X,N)
C
      DC 1 I=1,N
    1 XS(I,1) = SIGXZ(I)*X(I)+XZMEAN(I)
C
      RETURN
      END
C
C
C
      SUBROUTINE ADD (A,B,N,M,C)
C
C     THIS SUBROUTINE ADDS THE NXM MATRICES A AND B, STORING THE
C     RESULT IN C
      REAL *8A,B,C
      DIMENSION A(9,9), B(9,9), C(9,9)
C
      DC 1 I=1,N
C
```

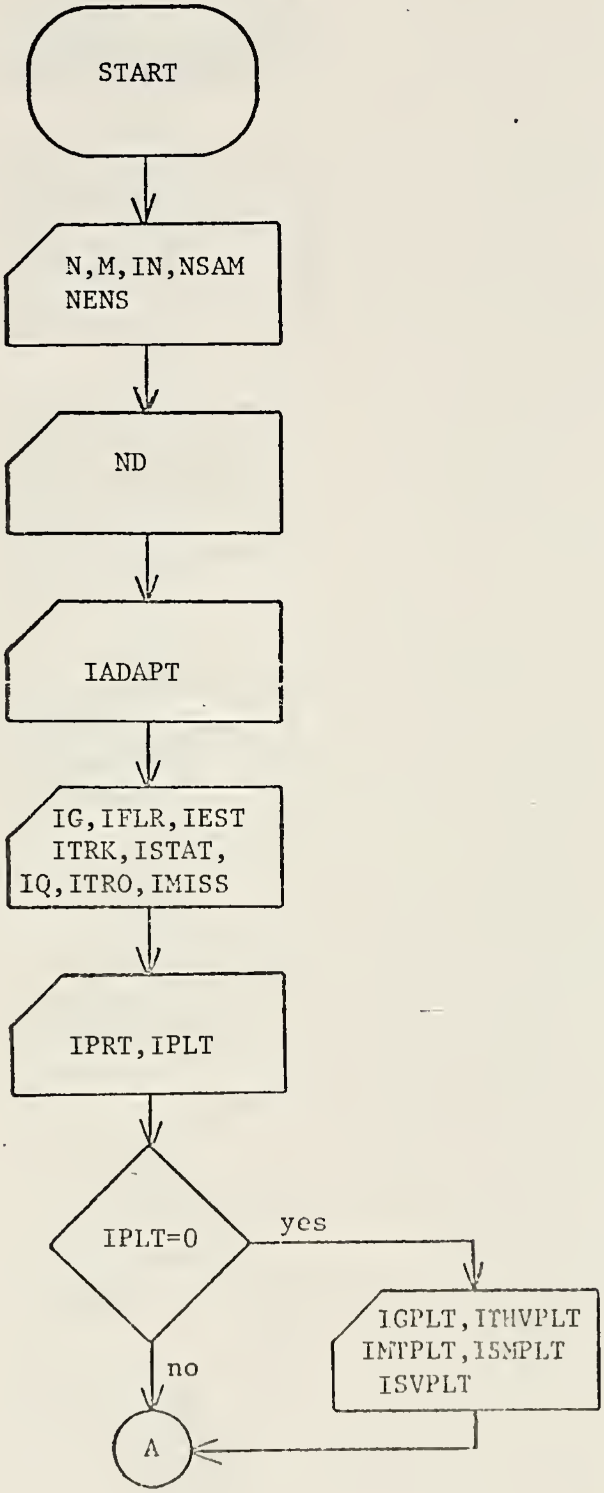
```
      DO 1 J=1,M
    1 C(I,J) = A(I,J)+B(I,J)
C
      RETURN
      END
C
C
C
      SUBROUTINE MREAD (A,N,M)
C
C     THIS SUBROUTINE READS AN NXM MATRIX A ACCORDING TO THE FORMAT
C     8D10.5. THE ENTRIES IN THE FIRST ROW OF A ARE READ FIRST, THEN
C     THE ENTRIES IN THE SECOND ROW, AND SO ON.
C
      REAL *8A
      DIMENSION A(9,9)
C
      DO 1 I=1,N
    1 READ (5,2) (A(I,J),J=1,M)
C
    2 FORMAT (8F10.0)
      RETURN
      END
C
C
C
      SUBROUTINE MWRITE (A,N,M)
C
C     THIS SUBROUTINE WRITES THE ENTRIES OF THE NXM MATRIX A
      REAL *8A
      DIMENSION A(9,9)
C
      DO 1 I=1,N
    1 WRITE (6,2) (A(I,J),J=1,M)
C
    2 FORMAT (9(2X,1PE12.5))
      RETURN
      END
C
C
C
      SUBROUTINE PROD (A,B,N,M,L,C)
C
C     THIS SUBROUTINE COMPUTES THE MATRIX PRODUCT AB AND STORES THE
C     RESULT IN C
C     A = NXM,    B = MXL,    C = NXL
      REAL *8A,B,C,T
      DIMENSION A(9,9), B(9,9), C(9,9), T(9,9)
```

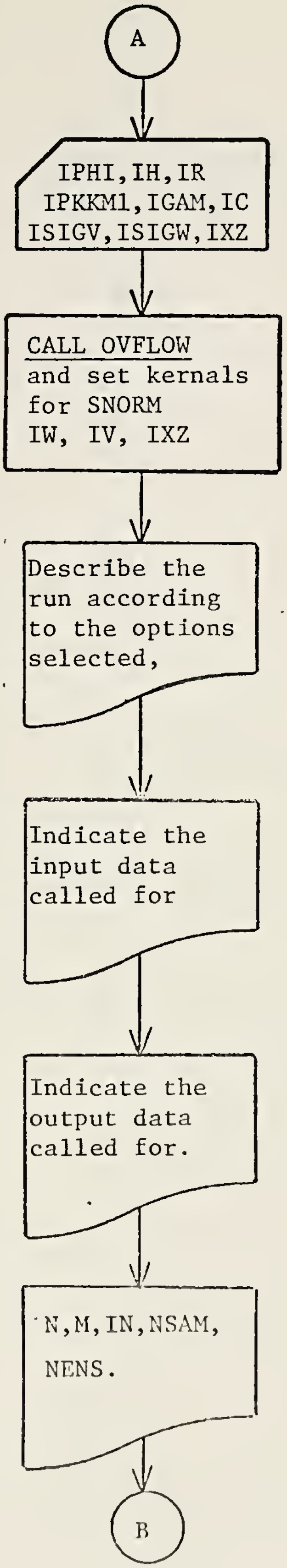
```
C
      DO 1 I=1,N
C
      DO 1 J=1,L
    1 T(I,J) = 0.0
C
C
      DO 2 I=1,N
C
      DO 2 J=1,L
C
      DO 2 K=1,M
    2 T(I,J) = T(I,J)+A(I,K)*B(K,J)
C
C
      DO 3 I=1,N
C
      DO 3 J=1,L
    3 C(I,J) = T(I,J)
C
      RETURN
      END
C
C
C
      SUBROUTINE SUB (A,B,N,M,C)
C
C     THIS SUBROUTINE SUBTRACTS THE NXM MATRIX B FROM THE NXM MATRIX
C     A AND STORES THE RESULT IN C
      REAL *8A,B,C
      DIMENSION A(9,9), B(9,9), C(9,9)
C
      DO 1 I=1,N
C
      DO 1 J=1,M
    1 C(I,J) = A(I,J)-B(I,J)
C
      RETURN
      END
C
C
C
      SUBROUTINE TRANS (A,N,M,C)
C
C     THIS SUBROUTINE FORMS THE MATRIX TRANSPOSE CF A STORING THE
C     RESULT IN C
C     A = NXM,   C = MXN
C
```

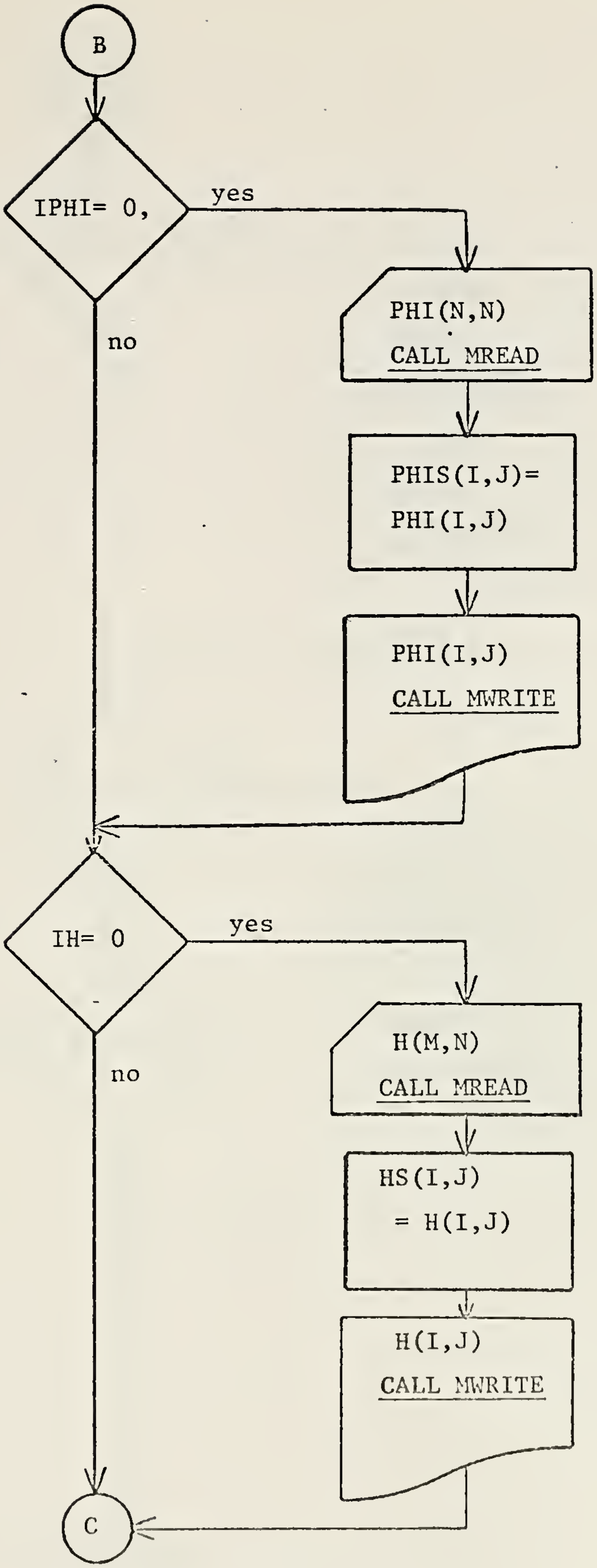
```
      REAL *8A,C
      DIMENSION A(9,9), C(9,9)
C
      DO 1 I=1,N
C
      DC 1 J=1,M
    1 C(J,I) = A(I,J)
C
      RETURN
      END
C
C
C
      SUBROUTINE VADD (X,Y,N,Z)
C
C     THIS SUBROUTINE COMPUTES THE SUM OF THE N-VECTORS X AND
C     Y AND STORES THE RESULT IN THE N-VECTOR Z
C
      REAL *4X(9),Y(9),Z(9)
C
      DC 1 I=1,N
    1 Z(I) = X(I)+Y(I)
C
      RETURN
      END
C
C
C
      SLBROUTINE VPROD (A,X,M,N,Y)
C
C     THIS SUBROUTINE COMPUTES THE PRODUCT OF THE MXN MATRIX
C     A AND THE N-VECTOR X AND STORES THE RESULT IN THE
C     M-VECTOR Y
C
C
      REAL *4A(9,9),X(9),Y(9),T(9)
C
      DO 1 I=1,M
      T(I) = 0.D0
C
      DO 1 J=1,N
    1 T(I) = T(I)+A(I,J)*X(J)
C
C
      DO 2 I=1,M
    2 Y(I) = T(I)
C
      RETURN
```

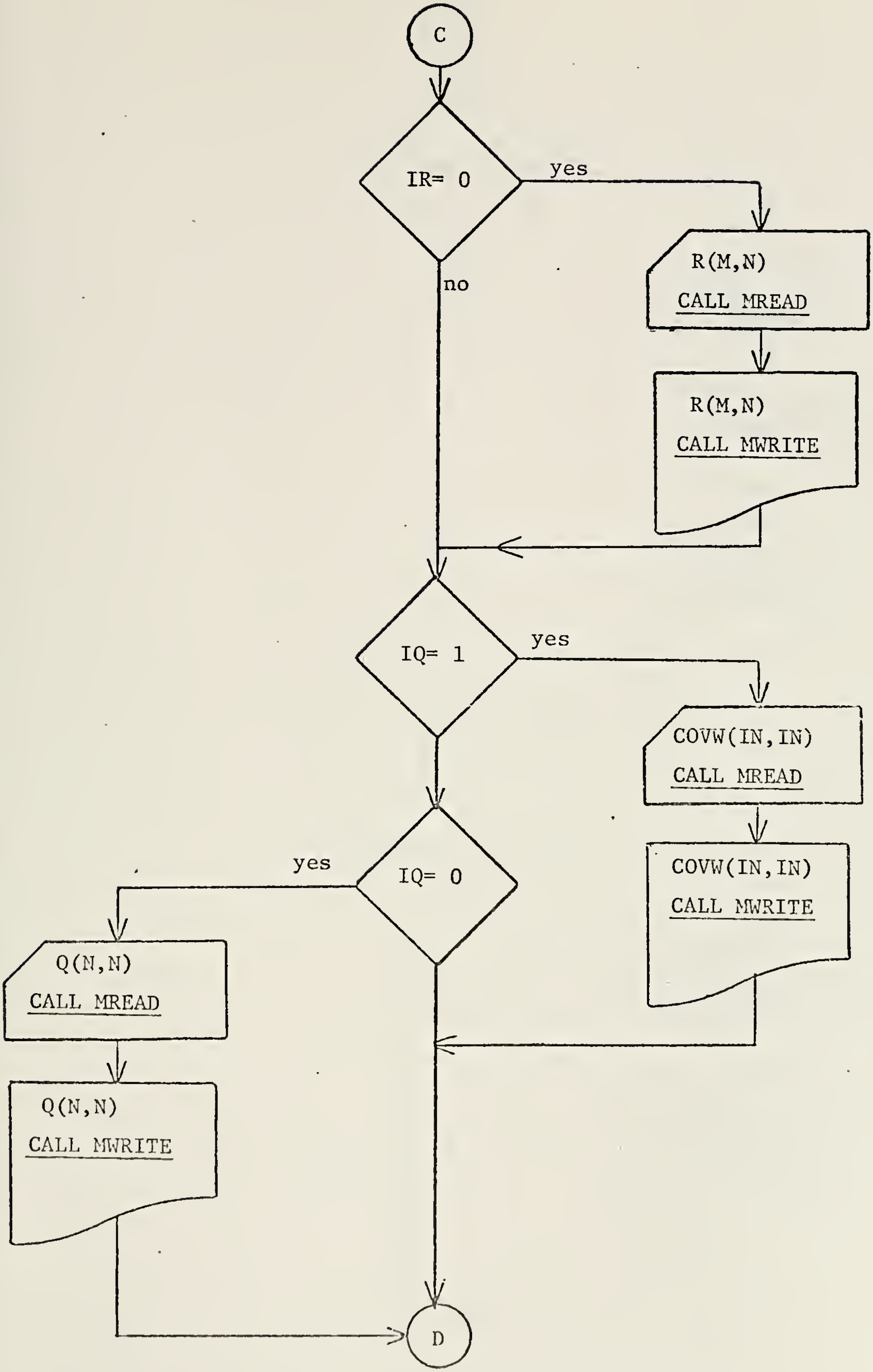
```
      END
C
C
C
      SUBROUTINE VREAD (V,N)
C
C     THIS SUBROUTINE READS THE N-DIMENSIONAL S.P. VECTOR V
C
      DIMENSION V(9)
      READ (5,1) (V(I),I=1,N)
    1 FORMAT (8F10.0)
      RETUFN
      END
C
C
C
      SUBROUTINE VSUB (X,Y,N,Z)
C
C     THIS SUBROUTINE COMPUTES THE DIFFERENCE X-Y OF THE TWO
C     N-VECTORS X & Y AND STORES THE RESULT IN THE N-VECTOR Z
C
      PEAL *4X(9),Y(9),Z(9)
C
      DO 1 I=1,N
    1 Z(I) = X(I)-Y(I)
C
      RETURN
      END
C
C
C
      SUBROUTINE VWRITE (V,N)
C
C     THIS SUBROUTINE WRITES THE N-DIMENSIONAL S.P. VECTOR V
C
      DIMENSION V(9)
      WRITE (6,1) (V(I),I=1,N)
    1 FORMAT (9(2X,1PE12.5))
      RETURN
      END
```

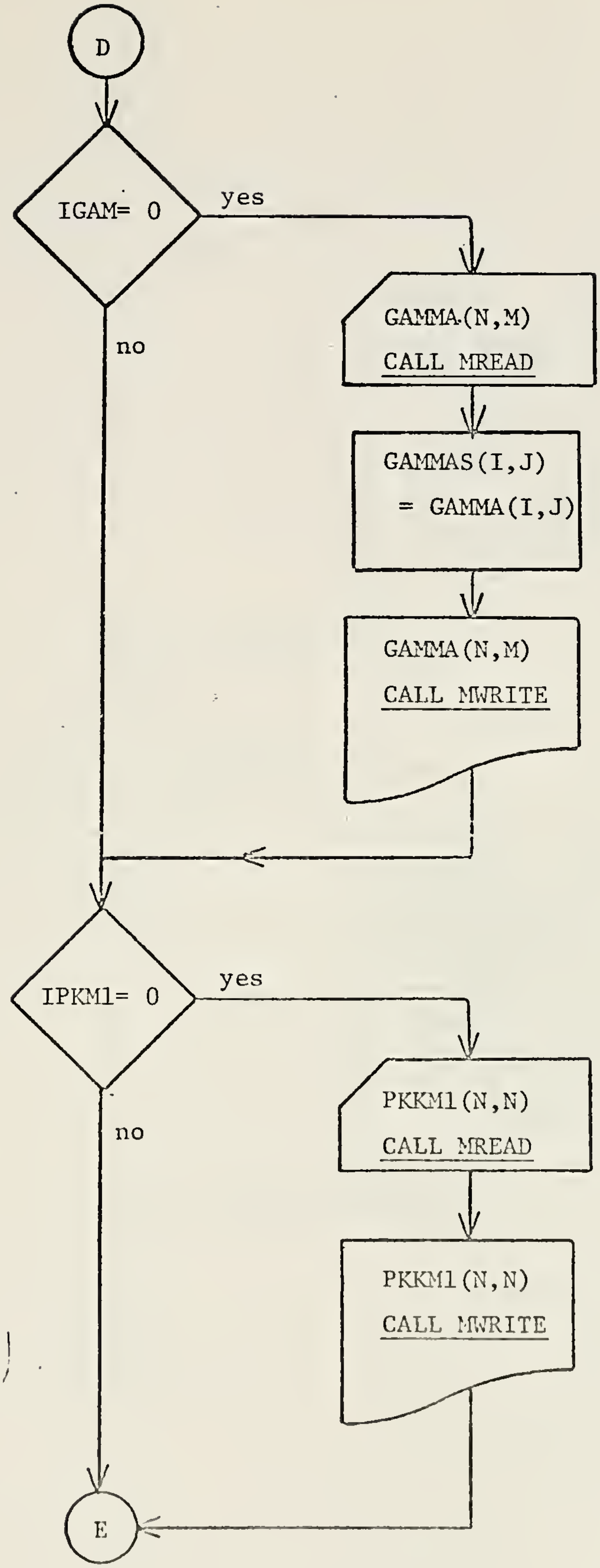
A.3 Flowchart of the Monte-Carlo Simulation Program

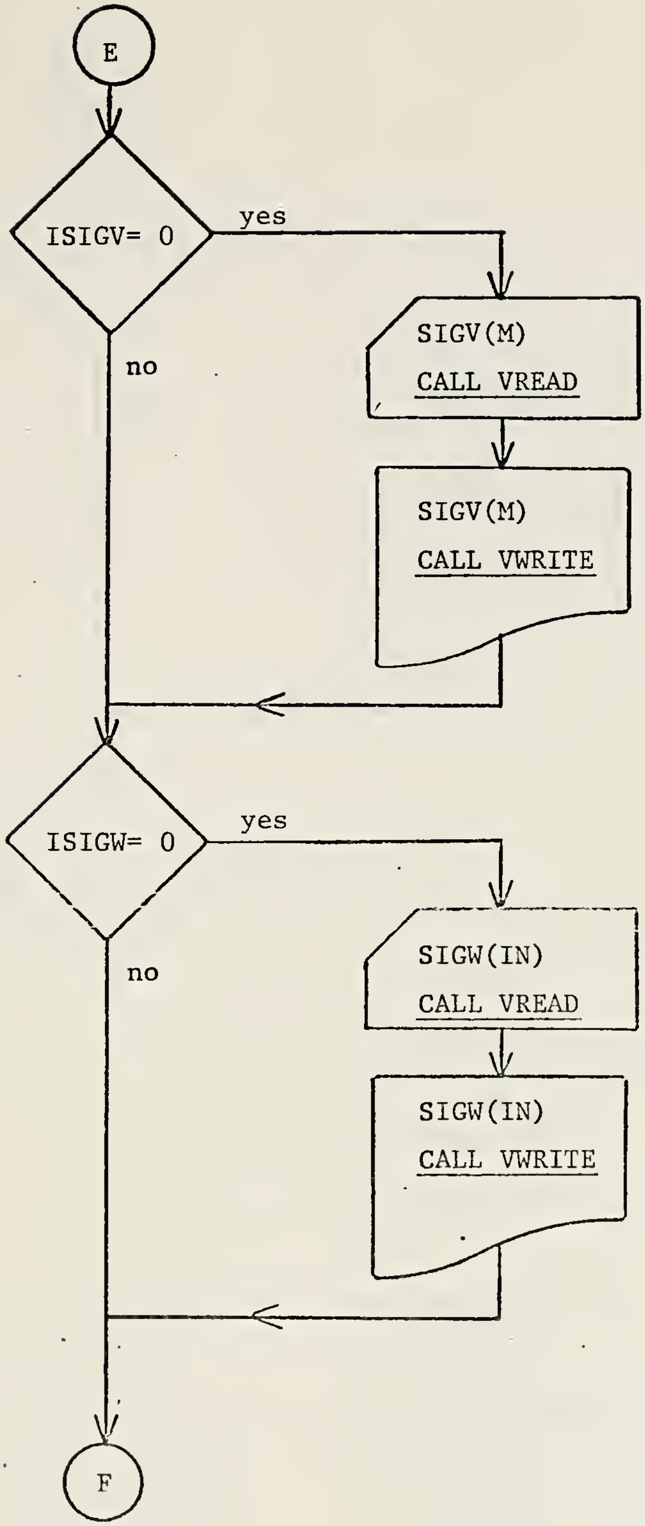


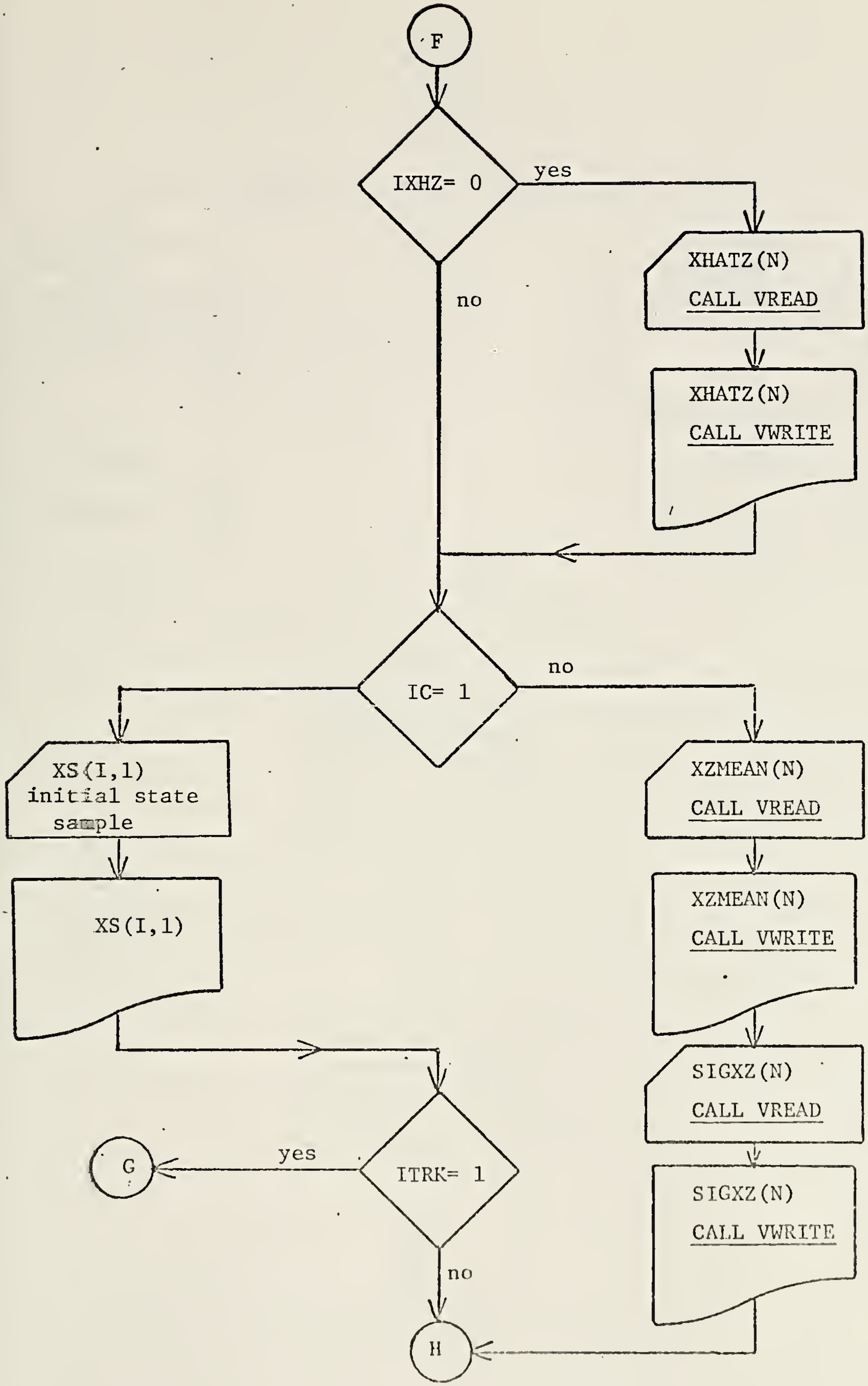

A
IPHI,IH,IR
IPKKM1,IGAM,IC
ISIGV,ISIGW,IXZ
CALL OVFLOW
and set kernals
for SNORM
IW, IV, IXZ
Describe the
run according
to the options
selected,
Indicate the
input data
called for
Indicate the
output data
called for.
N,M,IN,NSAM,
NENS.
B

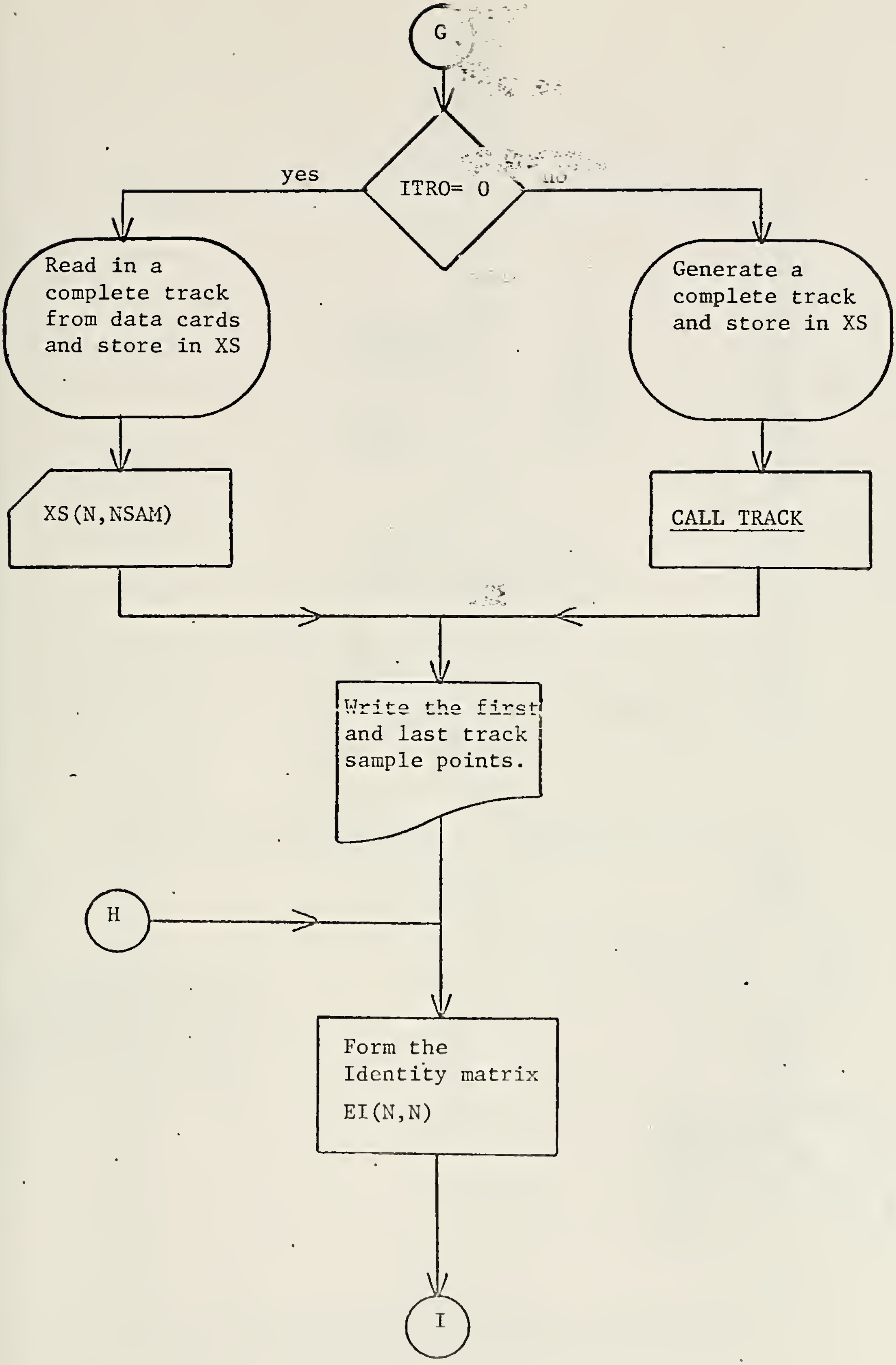

B
IPHI= 0,
yes
no
PHI(N,N)
CALL MREAD
PHIS(I,J)=
PHI(I,J)
PHI(I,J)
CALL MWRITE
IH= 0
yes
no
H(M,N)
CALL MREAD
HS(I,J)
= H(I,J)
H(I,J)
CALL MWRITE
C

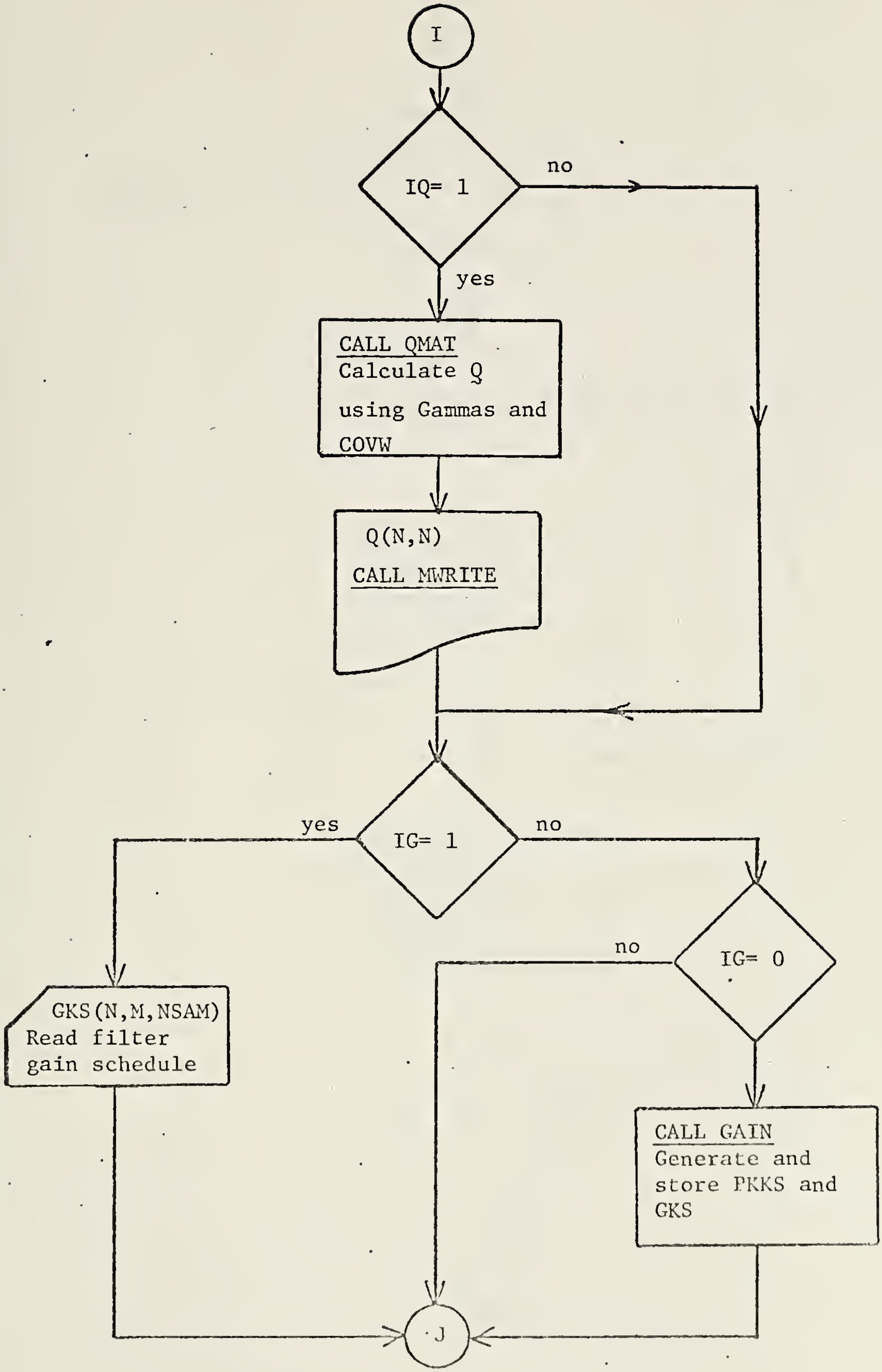

C
IR= 0
yes
no
R(M,N)
CALL MREAD
R(M,N)
CALL MWRITE
IQ= 1
yes
COVW(IN,IN)
CALL MREAD
COVW(IN,IN)
CALL MWRITE
IQ= 0
yes
Q(N,N)
CALL MREAD
Q(N,N)
CALL MWRITE
D

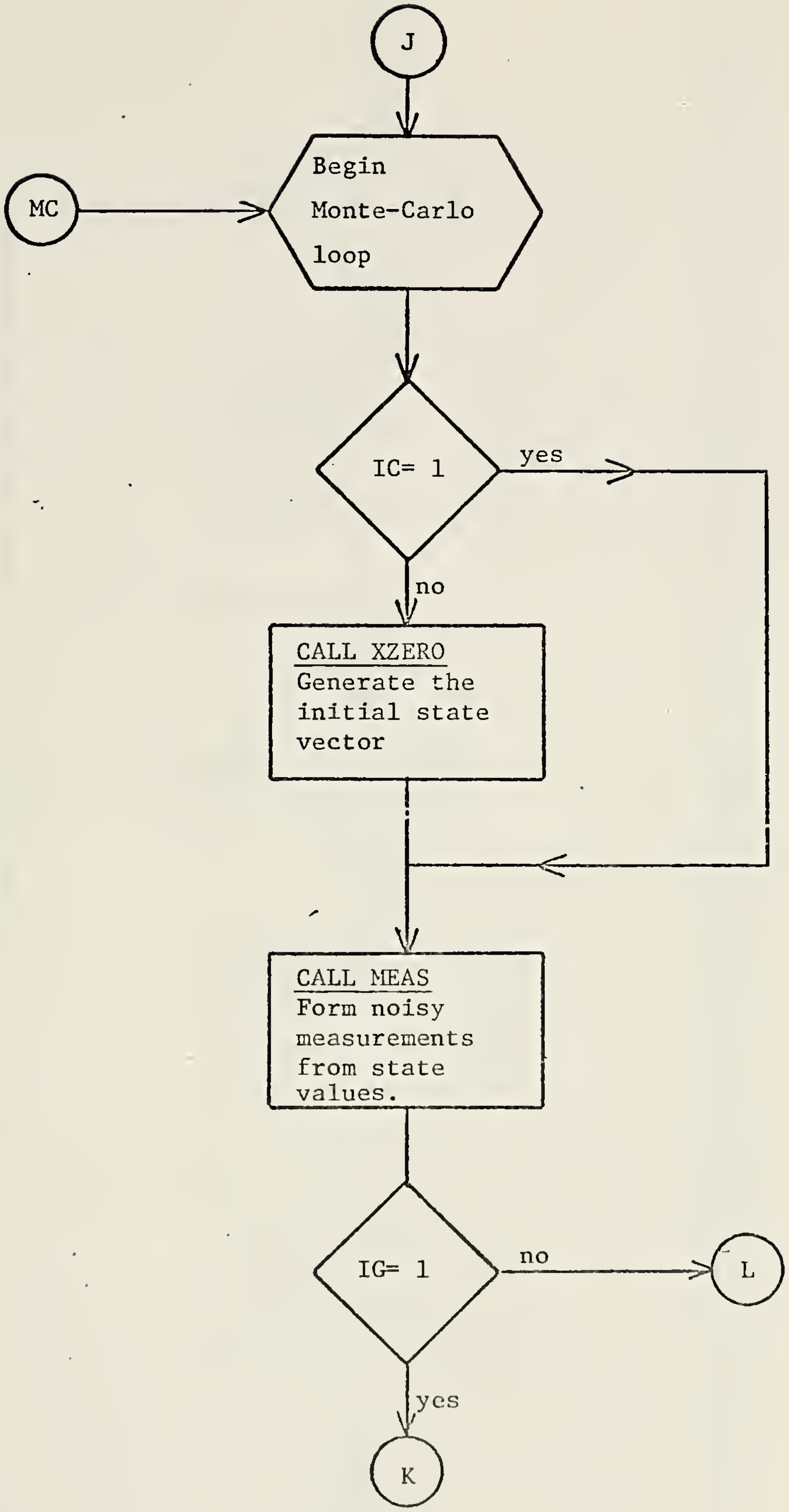

D
IGAM= 0
yes
no
GAMMA(N,M)
CALL MREAD
GAMMAS(I,J)
= GAMMA(I,J)
GAMMA(N,M)
CALL MWRITE
IPKM1= 0
yes
no
PKKM1(N,N)
CALL MREAD
PKKM1(N,N)
CALL MWRITE
E

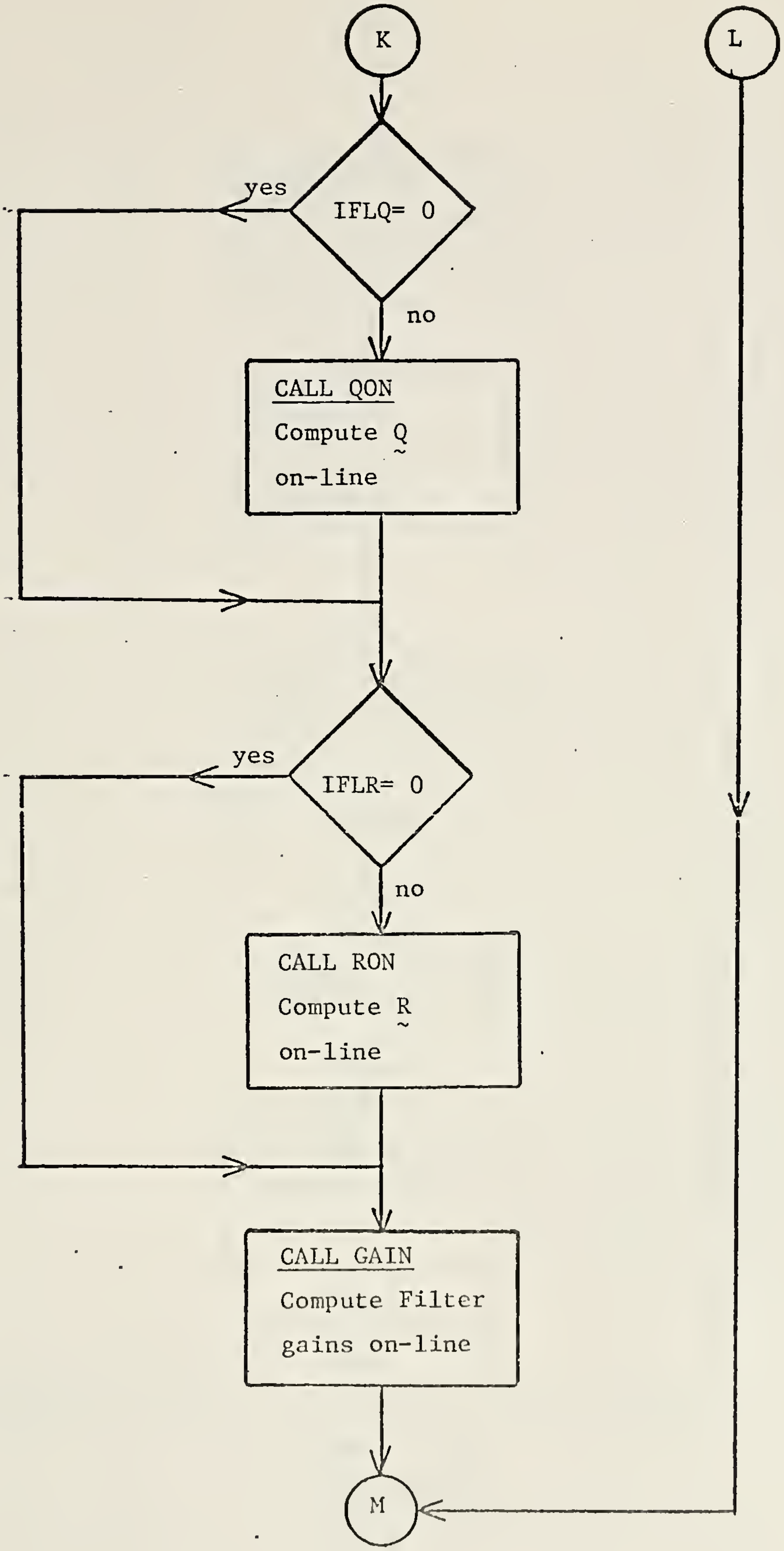

E
ISIGV= 0
yes
no
SIGV(M)
CALL VREAD
SIGV(M)
CALL VWRITE
ISIGW= 0
yes
no
SIGW(IN)
CALL VREAD
SIGW(IN)
CALL VWRITE
F

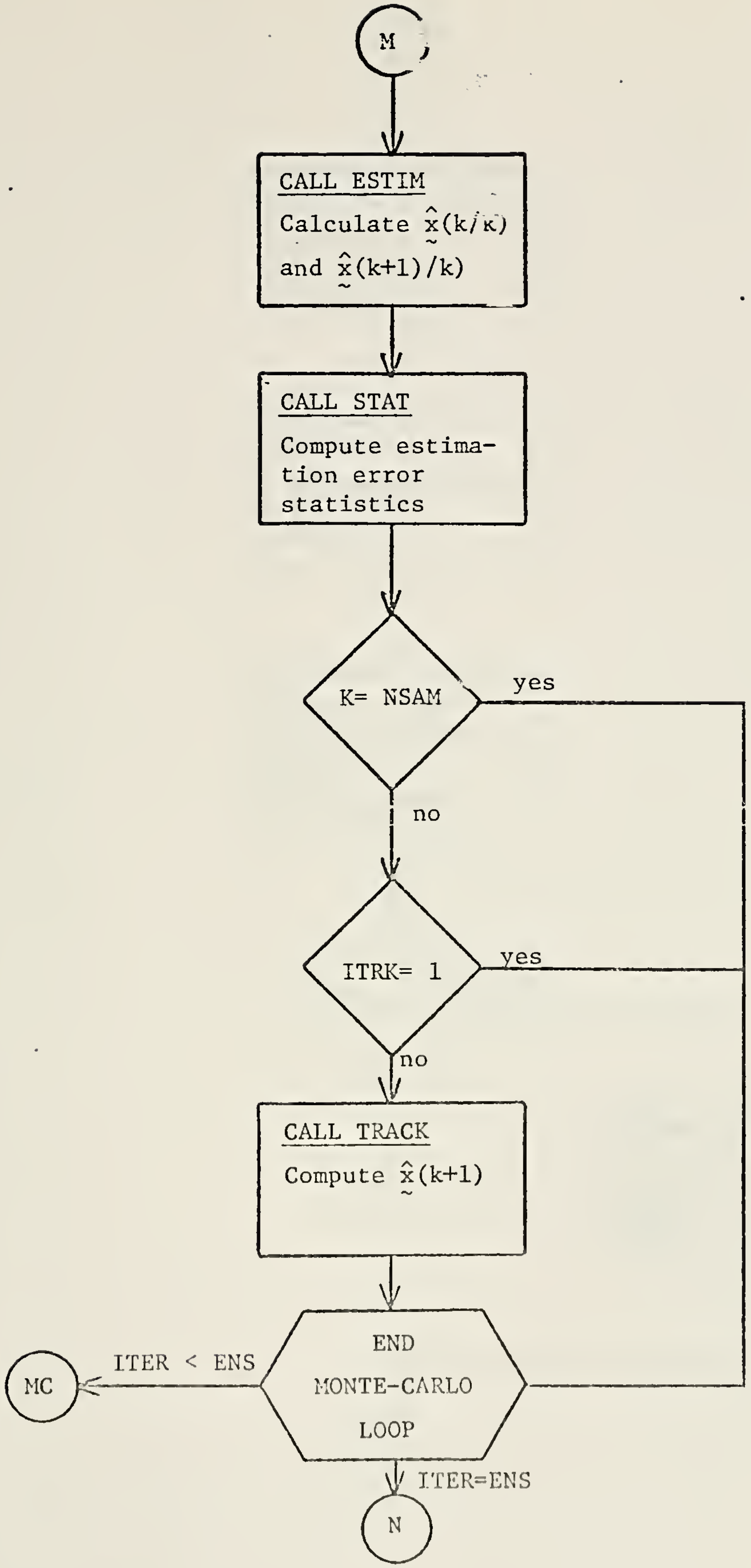

F
IXHZ= 0
yes
no
XHATZ(N)
CALL VREAD
XHATZ(N)
CALL VWRITE
IC= 1
no
XS(I,1)
initial state
sample
XS(I,1)
XZMEAN(N)
CALL VREAD
XZMEAN(N)
CALL VWRITE
SIGXZ(N)
CALL VREAD
SIGXZ(N)
CALL VWRITE
G
yes
ITRK= 1
no
H

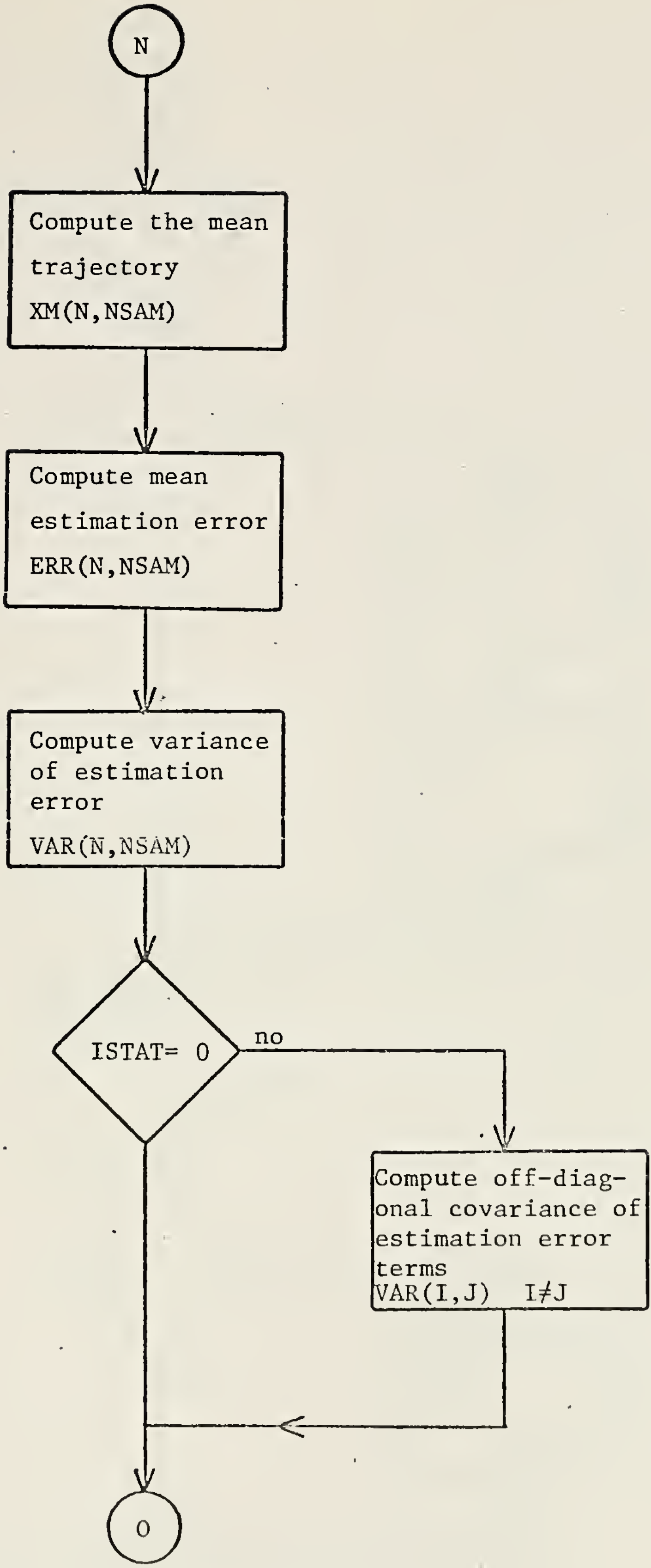

G
ITRO= 0
yes
no
Read in a complete track from data cards and store in XS
Generate a complete track and store in XS
XS(N,NSAM)
CALL TRACK
Write the first and last track sample points.
H
Form the Identity matrix
EI(N,N)
I

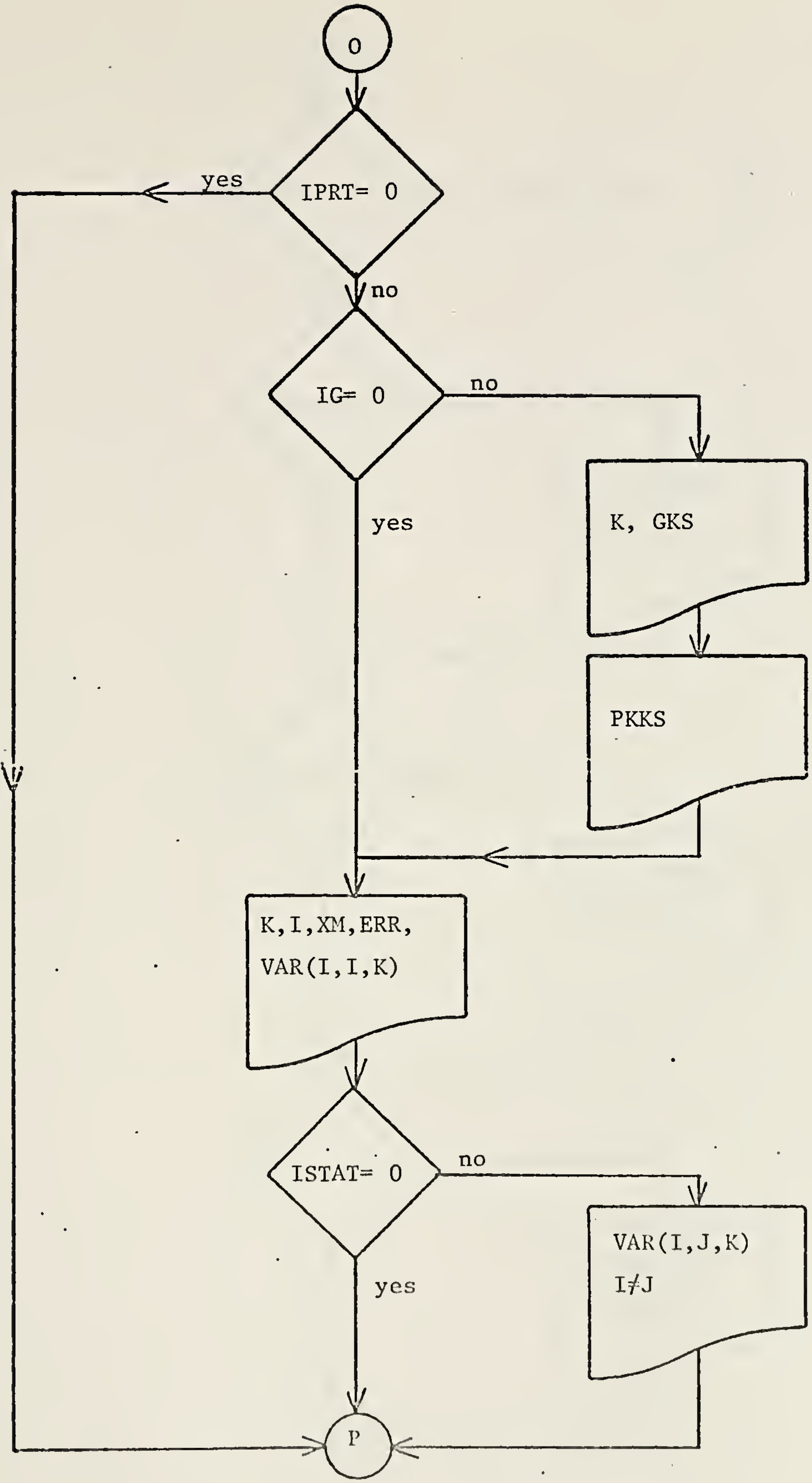

I
IQ= 1
no
yes
CALL QMAT
Calculate Q
using Gammas and
COVW
Q(N,N)
CALL MWRITE
yes
IG= 1
no
no
IG= 0
GKS(N,M,NSAM)
Read filter
gain schedule
CALL GAIN
Generate and
store PKKS and
GKS
J

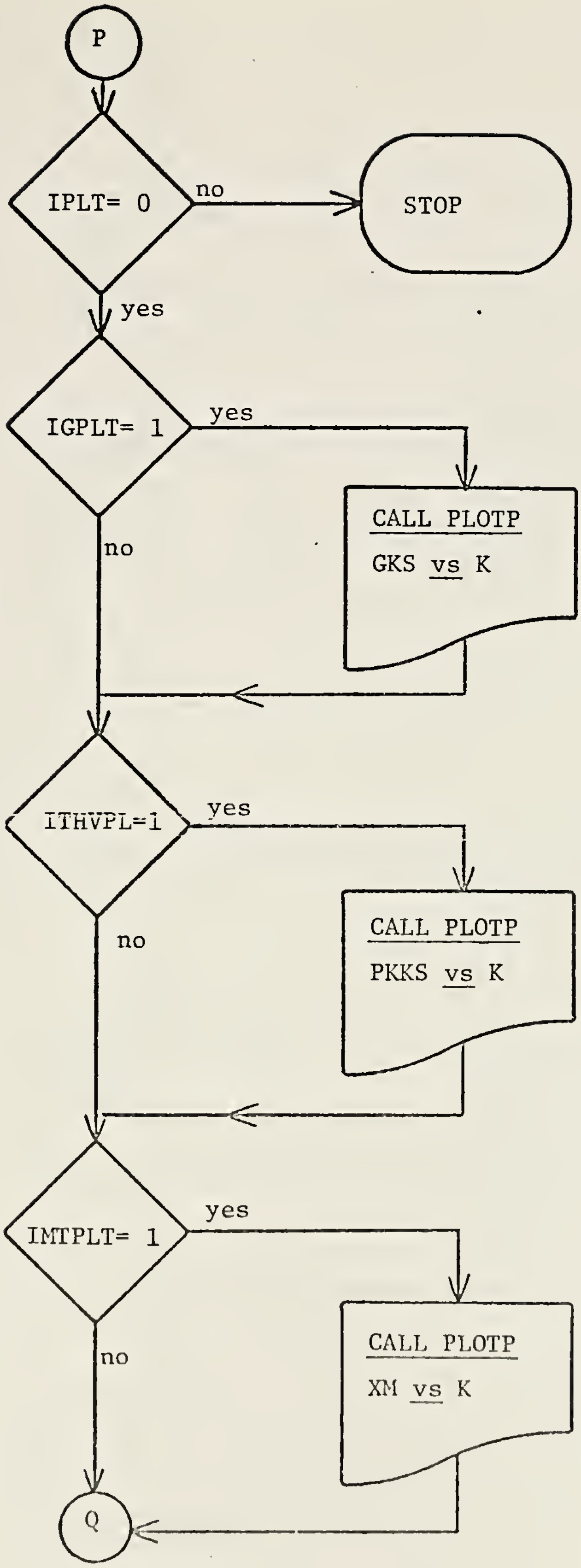

J
MC
Begin
Monte-Carlo
loop
IC= 1
yes
no
CALL XZERO
Generate the
initial state
vector
CALL MEAS
Form noisy
measurements
from state
values.
IG= 1
no
L
yes
K

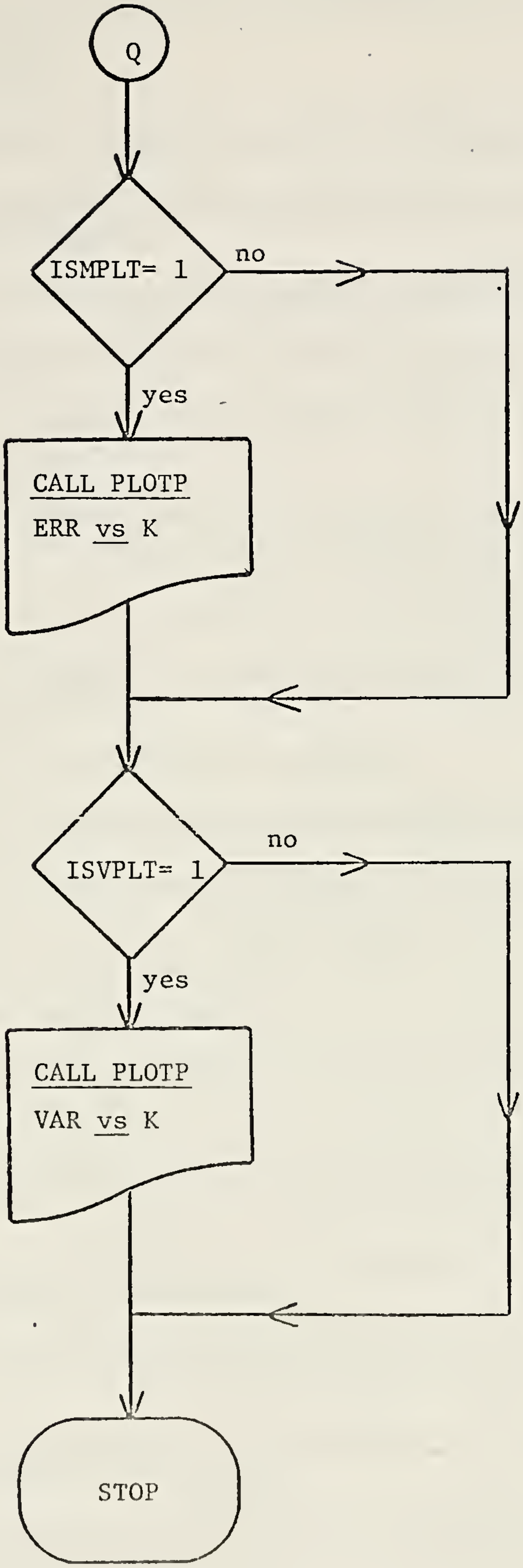

K
L
IFLQ= 0
yes
no
CALL QON
Compute Q
on-line
IFLR= 0
yes
no
CALL RON
Compute R
on-line
CALL GAIN
Compute Filter
gains on-line
M


M
CALL ESTIM
Calculate $\hat{\underset{\sim}{x}}(k/k)$
and $\hat{\underset{\sim}{x}}(k+1)/k)$
CALL STAT
Compute estimation error statistics
K= NSAM
yes
no
ITRK= 1
yes
no
CALL TRACK
Compute $\hat{\underset{\sim}{x}}(k+1)$
END MONTE-CARLO LOOP
ITER < ENS
MC
ITER=ENS
N


N
Compute the mean trajectory
XM(N,NSAM)
Compute mean estimation error
ERR(N,NSAM)
Compute variance of estimation error
VAR(N,NSAM)
ISTAT= 0
no
Compute off-diagonal covariance of estimation error terms
VAR(I,J) I≠J
O


O
IPRT= 0
yes
no
IG= 0
no
yes
K, GKS
PKKS
K,I,XM,ERR,
VAR(I,I,K)
ISTAT= 0
no
yes
VAR(I,J,K)
I≠J
P


P
IPLT= 0
no
STOP
yes
IGPLT= 1
yes
CALL PLOTP
GKS vs K
no
ITHVPL=1
yes
CALL PLOTP
PKKS vs K
no
IMTPLT= 1
yes
CALL PLOTP
XM vs K
no
Q


Q
ISMPLT= 1
no
yes
CALL PLOTP
ERR vs K
ISVPLT= 1
no
yes
CALL PLOTP
VAR vs K
STOP

A.4 Alphabetical Listing of Key Variables Used in the Monte-Carlo Program

AMISS(NSAM) - sample prediction miss distance vector

COVW(M,M) - covariance of forcing inputs matrix (double precision)

EI(N,N) - identity matrix (double precision)

ERR(N,NSAM) - sample estimation error vector

G(N,M) - the sample filter gain matrix (double precision)

GAMMA(N,N) - the matrix which relates the forcing inputs to the state vector (double precision)

GAMMAS(N,N) - the same as GAMMA but single precision

GK(N,M) - the sample filter gain matrix (single precision)

GKS(N,M,NSAM) - the filter gain schedule

H(M,N) - the observation matrix (double precision)

HS(M,N) - the observation matrix (single precision)

ITER - the ensemble member index

K - the sample index

PHI(N,N) - the state transition matrix (double precision)

PHIS(N,N) - the state transition matrix (single precision)

PKK(N,N) - the covariance of estimation error matrix (double precision)

PKKM1(N,N) - the one step prediction error covariance matrix (double precision)

PKKS(N,N) - the covariance of estimation error matrix (single precision)

Q(N,N) - the state excitation covariance matrix (double-precision)

R(N,N) - the covariance of measurement noise matrix (double precision)

SIGV(N) - the measurement noise standard deviation vector

SIGW(N) - the random forcing input standard deviation vector

SIGXZ(N) - the initial sample point standard deviation

V(N) - the measurement noise vector

W(N) - the forcing input vector

X(N) - the state vector

XHKK(N) - the state estimate vector

XHKKM1(N) - the one step state prediction vector

XM(N,NSAM) - the mean state trajectory over the ensemble

XS(N,NSAM) - the state trajectory for the current member of the ensemble

XZMEAN(N) - the mean of the initial state sample

Z(N) - the measurement vector

A.5 A List of IBM System/360 Source Library Subroutines Used by the Monte-Carlo Program

GAUSS3 - matrix inversion by the method of Gauss

OVFLOW - utility routine required by SNORM

PLOTP - plot on printer-multiple curves

SNORM - normally distributed pseudo-random deviations (shuffled)

APPENDIX B

PREDICTION ACCURACY CALCULATION

The accuracy of the predictions which can be made from each sample estimate are of prime concern in a gun fire control system since the gun is aimed at a predicted target position. A relatively simple procedure for measuring the prediction accuracy over realistic time intervals is described below.

The time interval over which the predictions are made is based on a straight line approximation of the time of flight (TOF) versus range curve for a 5"/54 caliber projectile [6] as indicated in Fig. 4:

Measurement of the slope of the straight line approximation in Fig. 4 shows that

$$\text{TOF} \simeq 1.43 \times 10^{-3} \quad (\text{s/yd}) \times \text{Range (yds)} \qquad (1)$$

Taking the maximum effective range of the gun against an air target into consideration, the time interval over which predictions are made is limited to a maximum of ten seconds. This corresponds to a maximum effective range of approximately 7,000 yards.

A further approximation is made in order to simplify the matrix manipulations required. The TOF is truncated to an integer multiple of the sample period. The prediction is based on the equation

$$\underset{\sim}{\hat{x}}(k/k-N) = \underset{\sim}{\Phi}^{N} \underset{\sim}{\hat{x}}(k-N/k-N) \qquad (2)$$

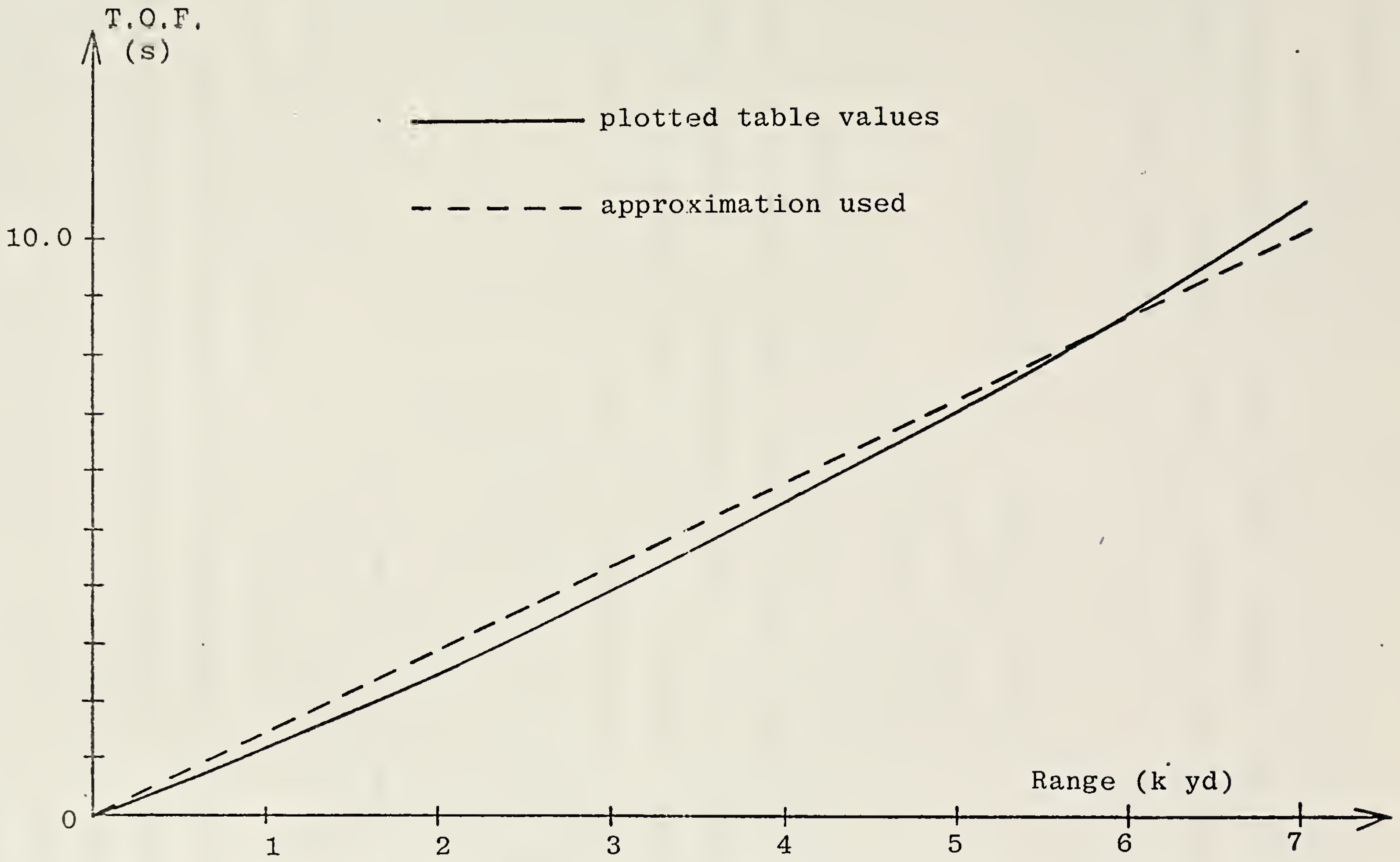


FIGURE 4 PLOT OF TIME OF FLIGHT VS. RANGE FOR A 5"/54 CAL. PROJECTILE

where N is the integer number of sample periods over which the prediction is made.

The prediction miss distance is calculated by finding the square root of the sum of the squares of the difference between the predicted position states and the respective true position states.

i.e.

$$\begin{aligned}(\text{miss distance})^2 &= [\hat{x}_1(k/k-N)-x_1(k)]^2\\ &+ [\hat{x}_2(k/k-N)-x_2(k)]^2\\ &+ [\hat{x}_3(k/k-N)-x_3(k)]^2 \qquad (3)\end{aligned}$$

The subroutine which calculates the prediction miss distance in the Monte-Carlo program stores a running sum of miss distances for each sample point over the ensemble so that the mean miss distance can be calculated. The subroutine also counts the number of predictions made (the number of samples which meet the projectile firing criteria), the number of predictions with miss distance less than 24 feet and the number of predictions with miss distances less than 48 feet for each run over the ensemble. The number of predictions with miss distance less than 48 feet at each sample point over the ensemble is also recorded.

The Fortran listing of subroutine Burst is included as part of the Monte-Carlo program in Appendix A. Notice that equation (2) is not evaluated in the Computer by raising the $n \times n$ matrix $\underset{\sim}{\Phi}$ to the N^{th} power by direct matrix algebra, which would be very time consuming, but is calculated by

operating on the individual components of $\underset{\sim}{\Phi}$. The N-step state transition matrix is given by

$$\underset{\sim}{\Phi}^{N} = \begin{bmatrix} \underset{\sim}{\Phi}_i & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\Phi}_i & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{\Phi}_i \end{bmatrix} \tag{4}$$

where for the constant-velocity model

$$\underset{\sim}{\Phi}_i = \begin{bmatrix} 1.0 & NT \\ 0.0 & 1.0 \end{bmatrix} \tag{5}$$

and for the correlated random acceleration model

$$\underset{\sim}{\Phi}_i = \begin{bmatrix} 1.0 & NT & \underset{\sim}{\Phi}_{13} \sum_{n=1}^{N} \underset{\sim}{\Phi}_{33}^{n-1} + T\underset{\sim}{\Phi}_{23} \sum_{n=1}^{N-1} [N-n]\underset{\sim}{\Phi}_{33}^{n-1} \\ 0.0 & 1.0 & \underset{\sim}{\Phi}_{23} \sum_{n=1}^{N} \underset{\sim}{\Phi}_{33}^{n-1} \\ 0.0 & 0.0 & \underset{\sim}{\Phi}_{33}^{N} \end{bmatrix} \tag{6}$$

where

$$\underset{\sim}{\Phi}_{13} = \frac{T}{\alpha} - \frac{1}{\alpha^2}\,[1-e^{-\alpha T}] \tag{7}$$

$$\underset{\sim}{\Phi}_{23} = \frac{1}{\alpha}\,[1-e^{-\alpha T}] \tag{8}$$

$$\underset{\sim}{\Phi}_{33} = e^{-\alpha T} \tag{9}$$

When the prediction is to be based on an assumption of constant velocity and the estimator is based on the model which assumes correlated random accelerations, $\underset{\sim}{\Phi}^{N}$ is calculated from equation (6) by setting $\underset{\sim}{\Phi}_{13}$, $\underset{\sim}{\Phi}_{23}$, and $\underset{\sim}{\Phi}_{33}$ equal to zero.

Equations (5) and (6) are derived by self-multiplying the matrices of equations (3.1) and (3.17) respectively and establishing a pattern.

APPENDIX C

A TARGET TRAJECTORY GENERATING PROGRAM

C.1 Description of the Program

In order to test the filter's performance against maneuvering targets it is desirable to have a realistic trajectory for which position, velocity and acceleration states are known at each sample point.

The maneuver decided upon is a periodic, planar maneuver described by the equations

$$y(x,t) = pA\sin^2(kx), \tag{1}$$

where

$$p = (-1)^n, \qquad n = \begin{matrix} 0, & kx < \pi \\ 1, & kx \geq \pi, \end{matrix}$$

and

$$k = \frac{2\pi}{\left[\begin{matrix}\text{the distance traveled along the x-axis} \\ \text{during the maneuver}\end{matrix}\right]}$$

$$\dot{y}(x,\dot{x},t) = 2pAk\dot{x}\cos(kx)\sin(kx) \tag{2}$$

and

$$\ddot{y}(x,\dot{x},\ddot{x},t) = 2pAk\{\ddot{x}\cos(kx)\sin(kx) + k\dot{x}^2[2\cos^2(kx)-1]\} \tag{3}$$

The form of the maneuver is shown in Fig. 5. The variables x, $\dot{x}$, and $\ddot{x}$ have been used to represent $x(t)$, $\dot{x}(t)$ and $\ddot{x}(t)$ respectively in equations (1) through (14) to make the notation less awkward.

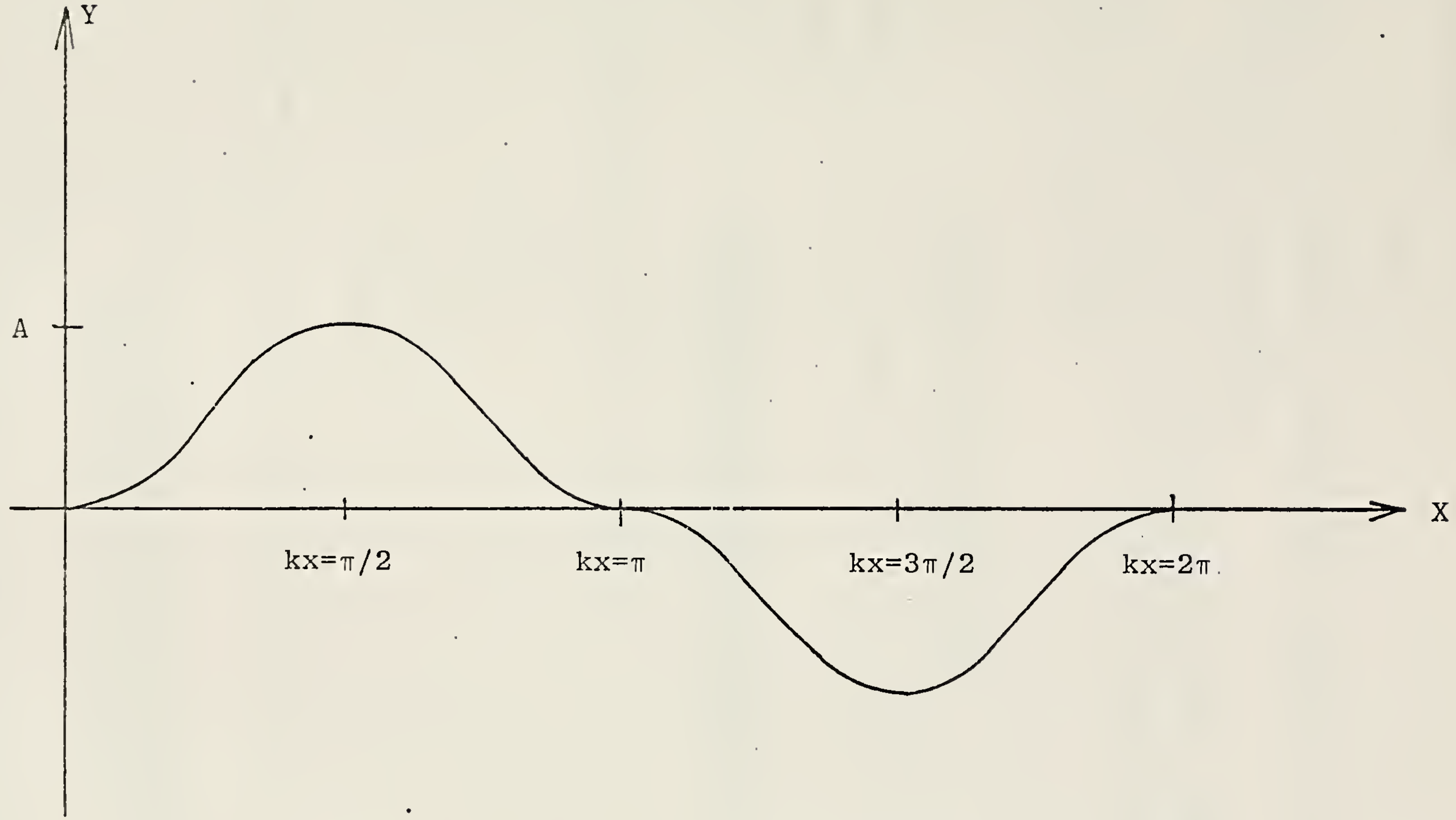


FIGURE 5 THE PLANAR MANEUVER

The two dimensional geometry of the maneuver yields the relationship

$$v^2 = \dot{x}^2 + \dot{y}^2 \tag{4}$$

where v, the speed of the aircraft tangent to the x-y plane, is assumed to be constant.

Transposing the terms of equation (4), substituting the right side of equation (2) for $\dot{y}$, and taking the square root of both sides yields

$$\dot{x} = \frac{v}{[1+4A^2k^2\cos^2(kx)\sin^2(kx)]^{\frac{1}{2}}} \, . \tag{5}$$

Equation (5) is differentiated with respect to time with the result

$$\ddot{x} = \frac{4A^2k^2v\dot{x}\sin(kx)\cos(kx)[1-2\cos^2(kx)]}{[1+4A^2k^2\cos^2(kx)\sin^2(kx)]^{3/2}} \tag{6}$$

If the system states are defined as

$$x_1 = x(t) \tag{7}$$

$$x_2 = \dot{x}(t) \tag{8}$$

$$x_3 = y(x,t) \tag{9}$$

$$x_4 = \dot{y}(x,\dot{x},t) \tag{10}$$

the resulting nonlinear state equations are

$$\dot{x}_1 = x_2 \tag{11}$$

$$\dot{x}_2 = \frac{4A^2k^2vx_2\sin(kx_1)\cos(kx_1)[1-2\cos^2(kx_1)]}{[1+4A^2k^2\cos^2(kx_1)\sin^2(kx_1)]^{3/2}} \tag{12}$$

$$\dot{x}_3 = x_4 \tag{13}$$

$$\dot{x}_4 = 2pAk\{\dot{x}_2\cos(kx_1)\sin(kx_1)-kx_2^2[1-2\cos^2(kx_1)]\} \tag{14}$$

The state equations are used in a computer program to make up the maneuvering portion of the trajectory. The computer program utilizes subroutine RKLDEQ, a Runge-Kutta-Gill differential equation solver, which is a member of the IBM System 360 Source Library.

In order that filter's performance against a target which maneuvers after the filter gains have reached steady state can be studied, provision is made for a constant-velocity section of target trajectory preceding the maneuver. A constant-velocity section of track following the maneuvers allows the filter's recovery time to be examined.

Although the trajectory is originally generated with the x-axis as its axis of symmetry, it may be rotated to any bearing angle and any first quadrant elevation angle. The geometry involved in the rotation of the axis of symmetry is illustrated in Fig. 6 through Fig. 8. As indicated in Fig. 6 for the elevation angle rotation, a new coordinate system is established which places the axis of symmetry, now labeled the x'-axis, at the desired elevation angle, E. The x and z-coordinate states can be calculated from the x' coordinate states.

$$x(k) = x'(k)\cos(E) \tag{15}$$

$$\dot{x}(k) = \dot{x}'(k)\cos(E) \tag{16}$$

$$\ddot{x}(k) = \ddot{x}'(k)\cos(E) \tag{17}$$

$$z(k) = x'(k)\sin(E) \tag{18}$$

$$\dot{z}(k) = \dot{x}'(k)\sin(E) \tag{19}$$

$$\ddot{z}(k) = \ddot{x}'(k)\sin(E) \tag{20}$$

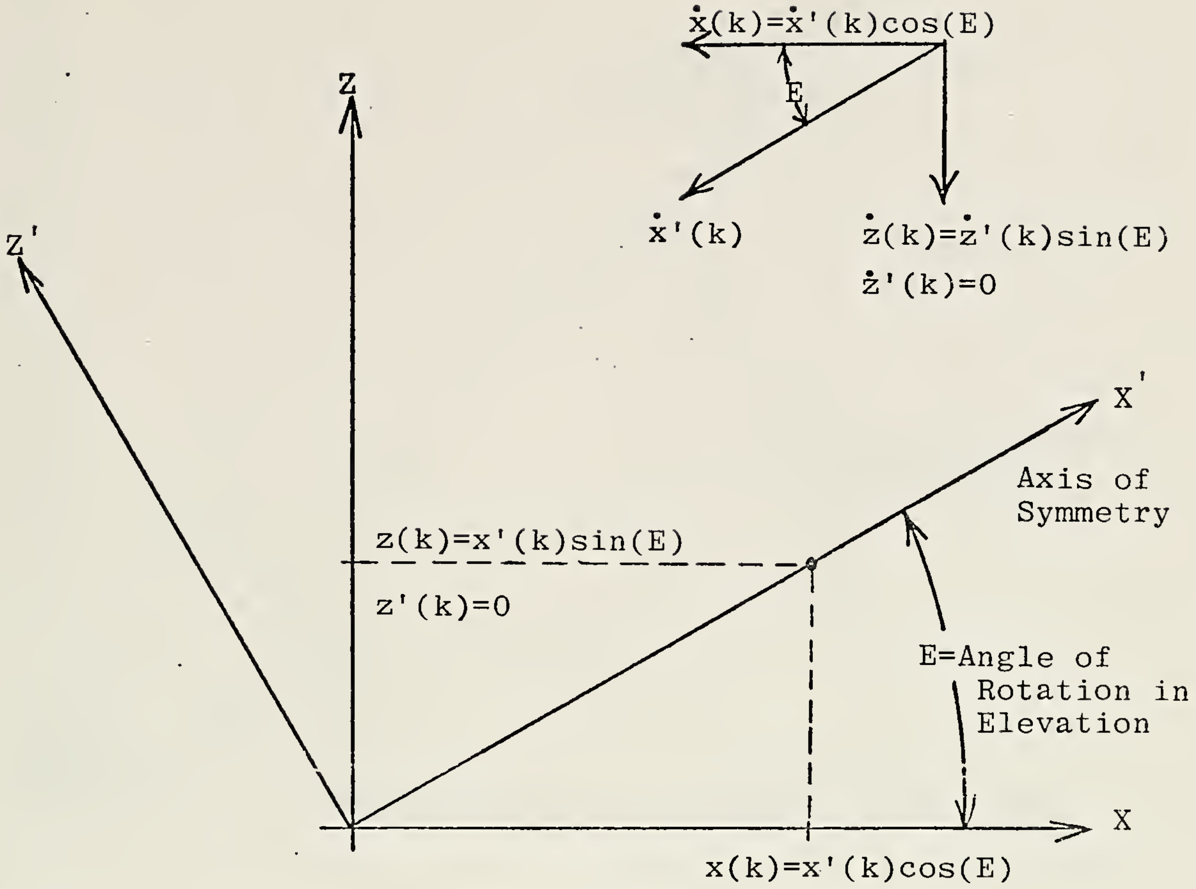


FIGURE 6 ROTATION OF THE TRACK IN ELEVATION (Vertical Plane)

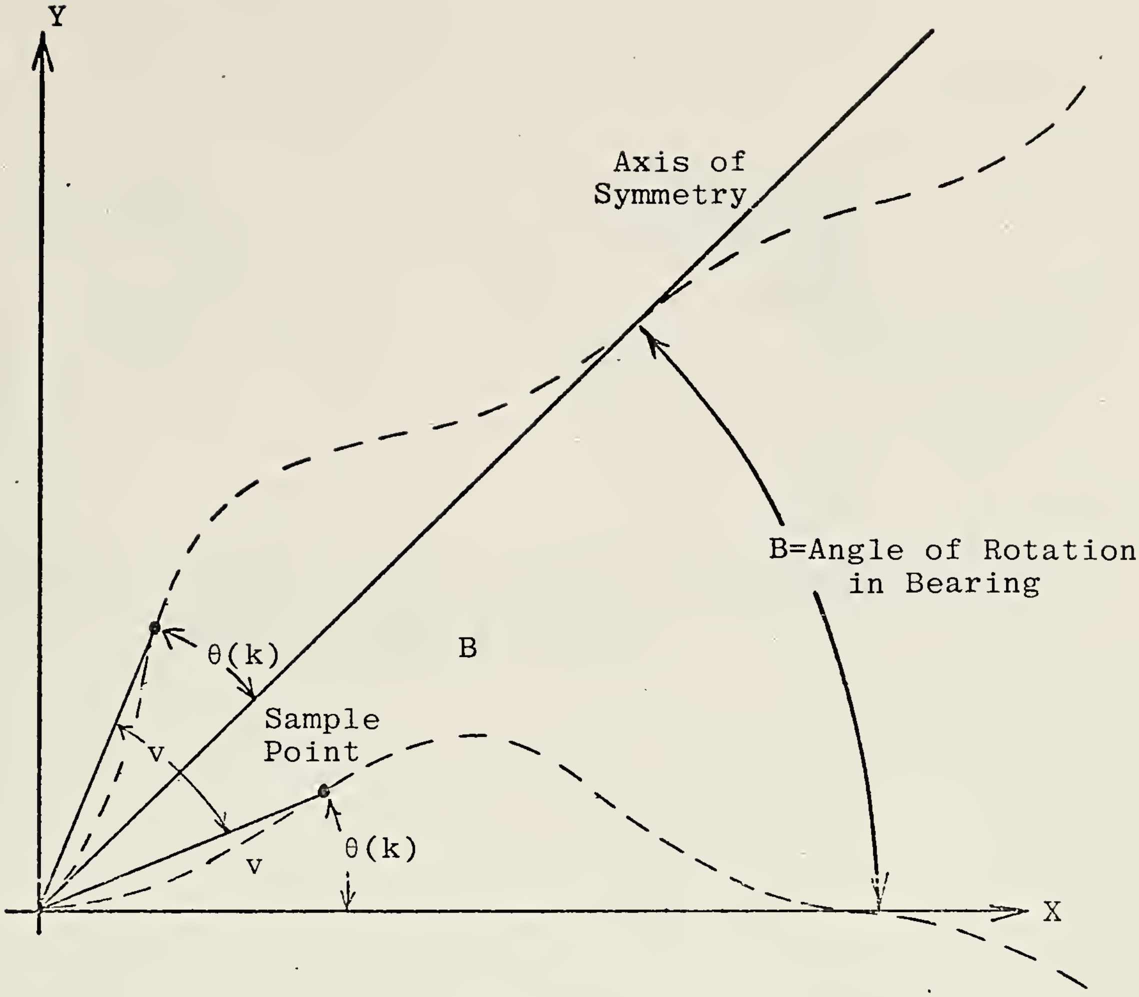


FIGURE 7 ROTATION OF THE TRACK AXIS IN BEARING

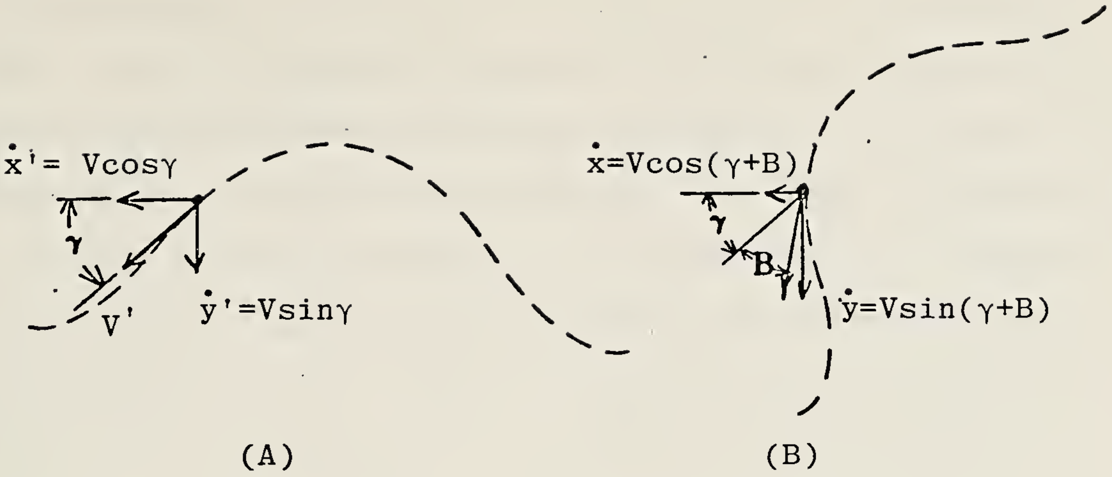


FIGURE 8 VELOCITY VECTOR BEFORE (A) AND AFTER (B) ROTATION THROUGH ANGLE B

The y-coordinate states are not affected by the elevation angle rotation since the rotation is about the y-axis. The bearing angle rotation is depicted in Fig. 7 -- the case shown is for a zero elevation angle. The axis of symmetry is rotated through the angle B, which results in a point on the trajectory curve being displaced from the x-axis by a total angle of $B + \theta$. Figure 7 shows the relationship between the x- and y-direction velocities before and after rotation. The z-coordinate states are not affected by the bearing rotation since the rotation is about the z-axis. The values of the x and y-coordinate states after rotation of the trajectory's axis of symmetry through elevation angle E and bearing angle B are given by

$$x(k) = R(k)\cos[B+\theta(k)]\cos(E) \ , \tag{21}$$

$$\dot{x}(k) = [\dot{x}'(k)\cos(B)-\dot{y}'(k)\sin(B)]\cos(E) \ , \tag{22}$$

$$\ddot{x}(k) = [\ddot{x}'(k)\cos(B)-\ddot{y}'(k)\sin(B)]\cos(E) \ , \tag{23}$$

$$y(k) = R(k)\sin[B+\theta(k)]\cos(E) \ , \tag{24}$$

$$\dot{y}(k) = \dot{x}'(k)\cos E\sin B+\dot{y}'(k)\cos B \ , \tag{25}$$

$$\ddot{y}(k) = \ddot{x}'(k)\cos E\sin B+\ddot{y}'\cos B \ , \tag{26}$$

where

$$R(k) = \sqrt{x'(k)^2+y'(k)^2} = \sqrt{x(k)^2+y(k)^2+z(k)^2} \tag{27}$$

and

$$\theta(k) = \arctan\ [y'(k)/x'(k)] \tag{28}$$

Equations (22) and (25) are derived from the trigonometric relationship shown in Fig. 7. Equations (23) and (26) are derived from a similar relationship between the acceleration states.

Provision is made for the trajectory to have non-zero intercepts with the y and/or z axis. This is accomplished simply by adding any desired value to the y-and/or z-position states at each sample point.

A subroutine is provided which performs conversion of all state values from cartesian to spherical coordinate quantities for cases in which spherical coordinate data is required. The spherical coordinate data is in units of yards in range and radians in bearing and elevation.

C.2 Relationship Between Velocity, Period of Maneuver, Amplitude of Maneuver, and Acceleration

The magnitude of the total acceleration for the x-y plane maneuver is given by

$$a = \sqrt{\ddot{x}^2 + \ddot{y}^2} \tag{29}$$

where $\ddot{x}$ and $\ddot{y}$ are given by equations (6) and (3) respectively. Making the substitutions as indicated in equation (29) and laboriously solving for a yields

$$a = \frac{2Ak^2v^2[1-2\cos^2(k)]}{1+4A^2k^2\cos^2(kx)\sin^2(kx)} \tag{30}$$

which can be further simplified by utilizing the trigonometric relationship

$$\cos 2\alpha = 2\cos^2\alpha - 1 \tag{31}$$

and

$$\sin 2\alpha = 2\sin\alpha\cos\alpha \tag{32}$$

with the result

$$a = \frac{-2Ak^2v^2\cos(2kx)}{1 + A^2k^2\sin^2(2kx)} \qquad (33)$$

To find the point, with respect to x-position, at which the maximum acceleration occurs during the maneuver it is necessary to differentiate equation (33) with respect to x and set the result equal to zero. This yields

$$4A^2k^2v^2\sin(2kx)[1+4Ak^2\sin^2(2kx)+Ak^2\cos^2(2kx)] = 0 \qquad (34)$$

which is true when

$$\sin(2kx) = 0,$$

$$\text{i.e. } 2kx=n\pi, \quad n=0,1,2,\ldots.$$

Substitution of $2kx = n\pi$ into equation (28) yields

$$|a|_{max} = 2Ak^2v^2 \qquad (35)$$

Equation (35) gives the relationship between target speed, maneuver amplitude, maneuver period, and the maximum acceleration required to complete the maneuver. This relationship is useful as an aid in choosing values for the variables involved.

C.3 Examples of Track Generating Program Use

Example 1. A trajectory is to be generated which approaches the origin from a range of 3,500 yards with a 20 degree dive angle at a constant speed of 200 yd/s. A 2.5 g maneuver of approximately ten seconds duration is to commence when the target is at a range of 2,700 yards. Sixty samples are to be generated at a four-hertz sampling rate.

The first task is to [illegible] the track specifications into the data required by the track generating program.

A maneuver duration of ten seconds is specified. To calculate the distance traveled along the axis of symmetry during a 2.5 g, ten second; $\sin^2$ maneuver would not be a simple task. However experience has shown that the change in range rate during such a maneuver is relatively small, therefore the distance is approximately

$$x_{per} = (10s)(200yd/s) = 2000\ yd. \tag{37}$$

Equation (36) can now be used to solve for the amplitude of the maneuver.

$$\begin{aligned} A &= \frac{2.5(g) \times 10.73(yd/s.g)}{2[\frac{2}{2000(yd)}]^2[200(yd/s)]^2} \\ &= 33.97\ yd. \end{aligned} \tag{38}$$

The four hertz sampling rate inverts to a 0.25 s sampling period.

The data is now available in the proper form to make up the data input cards for the track generating program. The cards must contain the following information.

Card 1:

S(1,1) = initial range = 3500.0
S(2,1) = initial range rate = -200.0

Card 2:

RATE = sampling period = 0.25
V = target speed = 200.0
A = maneuver amplitude = 33.97
XPER = maneuver duration = 2000.0
BR = bearing angle = 0.0 (no bearing specified)
E = elevation angle = 20.0
RMAN = maneuver commencement range = 2700.0
TRKEND = trajectory end range (the track is to be ended by the sample count) =-9999.9

Card 3:

CPA = y-direction displacement = 0.0
HO = z-direction displacement = 0.0

Card 4:

NSAM = no. of samples desired = 60
N = no. of states generated = 9
ICOORD = coordinate system indicator (data is to be in cartesian coordinates) = 0
IPUNCH = card output indicator (the output data is to be punched onto cards) = 1

Table VIII lists the data generated by the track generating program for this example.

Example 2. A level, constant-velocity trajectory beginning at the point x=5000 yards, y=1000 yards and z=333.3 yards in cartesian coordinates and proceeding in the negative x-direction with a speed of 200 yds/s until the x-direction displacement is 2800 yards is to be produced. The sampling period is 0.25 seconds. The output data is to be in spherical coordinates.

This trajectory can be generated quite simply by utilizing the CPA and HO features of the program. Since the trajectory is generated initially with the x-axis as its

TABLE VIII TRACK GENERATING PROGRAM OUTPUT FOR EXAMPLE 1

```
X-POSITION  X-VELOCITY  X-ACCELER    Y-POSITION  Y-VELOCITY  Y-ACCELER   Z-POSITION  Z-VELOCITY  Z-ACCELER
  3289.       -187.9     0.0          0.0         0.0         0.0          1197.      -68.40      0.0                  1
  3242.       -187.9     0.0          0.0         0.0         0.0          1180.      -68.40      0.0                  2
  3195.       -187.9     0.0          0.0         0.0         0.0          1163.      -68.40      0.0                  3
  3148.       -187.9     0.0          0.0         0.0         0.0          1146.      -68.40      0.0                  4
  3101.       -187.9     0.0          0.0         0.0         0.0          1129.      -68.40      0.0                  5
  3054.       -187.9     0.0          0.0         0.0         0.0          1112.      -68.40      0.0                  6
  3007.       -187.9     0.0          0.0         0.0         0.0          1094.      -68.40      0.0                  7
  2960.       -187.9     0.0          0.0         0.0         0.0          1077.      -68.40      0.0                  8
  2913.       -187.9     0.0          0.0         0.0         0.0          1060.      -68.40      0.0                  9
  2866.       -187.9     0.0          0.0         0.0         0.0          1043.      -68.40      0.0                 10
  2819.       -187.9     0.0          0.0         0.0         0.0          1026.      -68.40      0.0                 11
  2772.       -187.9     0.0          0.0         0.0         0.0          1009.      -68.40      0.0                 12
  2725.       -187.9     0.0          0.0         0.0         0.0          991.9      -68.40      0.0                 13
  2678.       -187.9     0.0          0.0         0.0         0.0          974.8      -68.40      0.0                 14
  2631.       -187.9     0.0          0.0         0.0         0.0          957.7      -68.40      0.0                 15
  2584.       -187.9     0.0          0.0         0.0         0.0          940.6      -68.40      0.0                 16
  2537.       -187.9     0.0          0.0         0.0         0.0          923.5      -68.40      0.0                 17
  2490.       -187.8     0.7887      -0.7599     -6.501      -25.45        906.4      -68.37      0.2871              18
  2443.       -187.6      1.269      -3.002      -12.42      -21.54        889.3      -68.27      0.4618              19
  2396.       -187.2      1.261      -6.499      -17.10      -15.56        872.2      -68.15      0.4591              20
  2350.       -187.0     0.7803      -10.90      -20.09      -8.188        855.2      -68.06      0.2840              21
  2303.       -186.9     0.11670-01  -15.78      -21.13      -0.1164       838.2      -68.02      0.42470-02          22
  2256.       -187.0    -0.7612      -20.67      -20.15       7.965        821.2      -68.05     -0.2771              23
  2209.       -187.2     -1.254      -25.10      -17.21       15.37        804.2      -68.15     -0.4563              24
  2163.       -187.6     -1.276      -28.63      -12.58       21.39        787.1      -68.27     -0.4543              25
  2116.       -187.8    -0.8078      -30.91      -6.682       25.38        770.0      -68.36     -0.2940              26
  2069.       -187.9    -0.24010-01  -31.71     -0.1011       26.82        752.9      -68.40     -0.17380-02          27
  2022.       -187.8     0.7693      -30.95       6.499       25.53        735.8      -68.37      0.2300              28
  1975.       -187.6      1.261      -28.71       12.45       21.68        718.8      -68.27      0.4591              29
  1928.       -187.3      1.269      -25.20       17.17       15.75        701.7      -68.15      0.4617              30
  1881.       -187.0     0.7991      -20.77       20.21       8.411        684.7      -68.06      0.2909              31
  1834.       -186.9     0.35010-01  -15.86       21.31      0.3491        667.7      -68.02      0.12740-01          32
  1788.       -187.0    -0.7419      -10.92       20.38      -7.742        650.7      -68.05     -0.2700              33
  1741.       -187.2     -1.246      -6.436       17.49      -15.17        633.6      -68.14     -0.4534              34
  1694.       -187.6     -1.282      -2.835       12.91      -21.25        616.6      -68.26     -0.4667              35
  1647.       -187.8    -0.8267     -0.4722       7.041      -25.30        599.5      -68.36     -0.3009              36
  1600.       -187.9    -0.48010-01  0.4176      0.4705      -26.82        582.4      -68.40     -0.17470-01          37
  1553.       -187.8     0.7496       1.167       6.454       25.60        565.3      -68.37      0.2728              38
  1506.       -187.6      1.254       3.404       12.43       21.82        548.2      -68.28      0.4563              39
  1459.       -187.3      1.275       6.912       17.19       15.95        531.2      -68.16      0.4642              40
  1413.       -187.0     0.8177       11.35       20.28       8.633        514.1      -68.06      0.2976              41
  1366.       -186.9     0.58330-01   16.29       21.44      0.5818        497.1      -68.02      0.2123D-01          42
  1319.       -187.0    -0.7224       21.26       20.57      -7.518        480.1      -68.05     -0.2629              43
  1272.       -187.2     -1.237       25.80       17.74      -14.98        463.1      -68.14     -0.4503              44
  1226.       -187.5     -1.288       29.46       13.19      -21.10        446.1      -68.26     -0.4689              45
  1179.       -187.8    -0.8453       31.90       7.357      -25.21        429.0      -68.36     -0.3077              46
  1132.       -187.9    -0.72000-01   32.86      0.7972      -26.81        411.9      -68.40     -0.2620D-01          47
  1085.       -187.9     0.7297       32.27      -5.820      -25.68        394.8      -68.37      0.2656              48
  1038.       -187.6      1.246       30.18      -11.82      -21.96        377.7      -68.28      0.4533              49
  990.9       -187.3      1.282       26.81      -16.62      -16.14        360.6      -68.16      0.4666              50
  944.1       -187.0     0.8360       22.50      -19.77      -8.854        343.6      -68.06      0.3043              51
  897.4       -186.9     0.81640-01   17.67      -20.99     -0.8145        326.6      -68.02      0.29710-01          52
  850.6       -187.0    -0.7026       12.80      -20.17       7.293        309.6      -68.05     -0.2557              53
  803.9       -187.2     -1.228       8.347      -17.39       14.78        292.6      -68.14     -0.4471              54
  757.0       -187.5     -1.294       4.759      -12.89       20.95        275.5      -68.26     -0.4710              55
  710.1       -187.8    -0.8637       2.392      -7.084       25.13        258.5      -68.36     -0.3143              56
  663.1       -187.9    -0.95960-01   1.489     -0.5363       26.80        241.4      -68.40     -0.34930-01          57
  616.1       -187.9     0.7095       2.146       6.089       25.74        224.3      -68.37      0.2583              58
  569.2       -187.9     0.0          0.0         0.0         0.0          207.2      -68.40      0.0                 59
  522.2       -187.9     0.0          0.0         0.0         0.0          190.1      -68.40      0.0                 60
```

axis of symmetry and the y and z-direction displacements are added afterward, the specified values of initial x-position and velocity are taken as the initial range and range-rate respectively. The other required input values are given directly in the track specifications.

The input data cards contain the following information.

Card 1:

S(1,1) = initial range (x-position) = 5000.0
S(2,1) = initial range rate (x-velocity) = -200.0

Card 2:

RATE = sampling period = 0.25
V = target speed = 200.0
A = maneuver amplitude = 0.0
XPER = maneuver duration = 0.0
BR = bearing angle = 0.0
E = elevation angle = 0.0
RMAN = maneuver commencement range = 0.0
TRKEND = trajectory end range = 2800.0

Card 3:

CPA = y-direction displacement = 1000.0
HO = z-direction displacement = 333.3

Card 4:

NSAM = number of samples = 200
(the track is range limited)
N = number of states generated = 9
ICOORD = coordinate system indicator = 1
(spherical coordinate data desired)
IPUNCH = card output indicator = 0
(no cards are desired)

Table IX lists the trajectory data generated by the track generating program for this example.

TABLE IX TRACK GENERATING PROGRAM OUTPUT FOR EXAMPLE 2

```
                                OUTPUT DATA IN SPHERICAL COORDINATES

RANGE     RDOT       R2DOT      BEARING      BDOT         B2DOT       ELEVATION      EDOT        E2DOT

5110.    -195.7      0.3331      0.1974    0.7692D-02   0.5917D-03   0.6528D-01   0.2504D-02   0.4703            1
5061.    -195.6      0.3429      0.1993    0.7842D-02   0.6089D-03   0.6591D-01   0.2551D-02   0.4786            2
5012.    -195.5      0.3530      0.2013    0.7997D-02   0.6267D-03   0.6655D-01   0.2600D-02   0.4872            3
4963.    -195.4      0.3635      0.2033    0.8156D-02   0.6452D-03   0.6721D-01   0.2651D-02   0.4959            4
4914.    -195.3      0.3745      0.2054    0.8319D-02   0.6644D-03   0.6788D-01   0.2702D-02   0.5049            5
4866.    -195.3      0.3859      0.2075    0.8488D-02   0.6844D-03   0.6856D-01   0.2756D-02   0.5141            6
4817.    -195.2      0.3977      0.2096    0.8662D-02   0.7052D-03   0.6926D-01   0.2811D-02   0.5235            7
4768.    -195.1      0.4100      0.2118    0.8841D-02   0.7269D-03   0.6997D-01   0.2867D-02   0.5332            8
4719.    -194.9      0.4229      0.2141    0.9025D-02   0.7494D-03   0.7069D-01   0.2925D-02   0.5432            9
4671.    -194.8      0.4362      0.2163    0.9216D-02   0.7728D-03   0.7143D-01   0.2985D-02   0.5534           10
4622.    -194.7      0.4502      0.2187    0.9412D-02   0.7972D-03   0.7218D-01   0.3047D-02   0.5636           11
4573.    -194.6      0.4647      0.2210    0.9614D-02   0.8227D-03   0.7295D-01   0.3110D-02   0.5746           12
4525.    -194.5      0.4798      0.2235    0.9823D-02   0.8492D-03   0.7374D-01   0.3176D-02   0.5856           13
4476.    -194.4      0.4957      0.2260    0.1004D-01   0.8768D-03   0.7454D-01   0.3243D-02   0.5969           14
4427.    -194.2      0.5121      0.2285    0.1026D-01   0.9056D-03   0.7536D-01   0.3313D-02   0.6085           15
4379.    -194.1      0.5294      0.2311    0.1049D-01   0.9357D-03   0.7620D-01   0.3385D-02   0.6204           16
4330.    -194.0      0.5474      0.2337    0.1073D-01   0.9670D-03   0.7705D-01   0.3459D-02   0.6327           17
4282.    -193.8      0.5662      0.2365    0.1098D-01   0.9998D-03   0.7793D-01   0.3535D-02   0.6452           18
4233.    -193.7      0.5858      0.2392    0.1123D-01   0.1034D-02   0.7882D-01   0.3614D-02   0.6582           19
4185.    -193.6      0.6064      0.2421    0.1149D-01   0.1070D-02   0.7974D-01   0.3696D-02   0.6714           20
4137.    -193.4      0.6279      0.2450    0.1176D-01   0.1107D-02   0.8067D-01   0.3780D-02   0.6850           21
4088.    -193.2      0.6504      0.2480    0.1205D-01   0.1146D-02   0.8163D-01   0.3867D-02   0.6990           22
4040.    -193.1      0.6741      0.2510    0.1234D-01   0.1187D-02   0.8260D-01   0.3957D-02   0.7134           23
3992.    -192.9      0.6988      0.2541    0.1264D-01   0.1230D-02   0.8360D-01   0.4050D-02   0.7281           24
3943.    -192.7      0.7247      0.2573    0.1295D-01   0.1275D-02   0.8463D-01   0.4146D-02   0.7433           25
3895.    -192.5      0.7519      0.2606    0.1328D-01   0.1322D-02   0.8568D-01   0.4245D-02   0.7588           26
3847.    -192.3      0.7805      0.2640    0.1361D-01   0.1372D-02   0.8675D-01   0.4348D-02   0.7748           27
3799.    -192.1      0.8105      0.2674    0.1396D-01   0.1423D-02   0.8785D-01   0.4455D-02   0.7912           28
3751.    -191.9      0.8420      0.2709    0.1433D-01   0.1478D-02   0.8898D-01   0.4565D-02   0.8081           29
3703.    -191.7      0.8752      0.2746    0.1470D-01   0.1535D-02   0.9013D-01   0.4679D-02   0.8254           30
3655.    -191.5      0.9100      0.2783    0.1509D-01   0.1595D-02   0.9132D-01   0.4798D-02   0.8431           31
3607.    -191.3      0.9467      0.2821    0.1550D-01   0.1658D-02   0.9253D-01   0.4920D-02   0.8614           32
3560.    -191.0      0.9854      0.2861    0.1592D-01   0.1724D-02   0.9378D-01   0.5048D-02   0.8801           33
3512.    -190.8       1.026      0.2901    0.1636D-01   0.1794D-02   0.9506D-01   0.5179D-02   0.8993           34
3464.    -190.5       1.069      0.2942    0.1682D-01   0.1867D-02   0.9637D-01   0.5316D-02   0.9190           35
3417.    -190.2       1.114      0.2985    0.1730D-01   0.1945D-02   0.9772D-01   0.5458D-02   0.9392           36
3369.    -190.0       1.162      0.3029    0.1779D-01   0.2026D-02   0.9910D-01   0.5606D-02   0.9599           37
3322.    -189.7       1.213      0.3074    0.1831D-01   0.2112D-02   0.1005       0.5759D-02   0.9811           38
3274.    -189.4       1.266      0.3120    0.1885D-01   0.2203D-02   0.1020       0.5918D-02    1.003           39
3227.    -189.0       1.323      0.3168    0.1941D-01   0.2299D-02   0.1035       0.6083D-02    1.025           40
3180.    -188.7       1.382      0.3218    0.2000D-01   0.2400D-02   0.1050       0.6255D-02    1.046           41
3133.    -188.3       1.446      0.3268    0.2061D-01   0.2507D-02   0.1066       0.6434D-02    1.071           42
3086.    -188.0       1.513      0.3321    0.2125D-01   0.2620D-02   0.1082       0.6619D-02    1.095           43
3039.    -187.6       1.584      0.3375    0.2192D-01   0.2740D-02   0.1099       0.6813D-02    1.119           44
2992.    -187.2       1.660      0.3430    0.2262D-01   0.2866D-02   0.1116       0.7014D-02    1.143           45
```

C.4 The Track Generating Program Listing

```
C        THE FOLLOWING PROGRAM GENERATES A SIMULATED AIRCRAFT
C        TRAJECTORY. THE TRAJECTORY MAY BE GENERATED WITH A MANEUVER IN
C        THE TRAVERSE PLANE BASED ON THE EQUATION    Y = A*SIN(X)**2
C        WHERE THE AMPLITUDE, PERIOD, AND COMMENCEMENT RANGE OF THE
C        MANEUVER MAY BE SELECTED BY THE USER. A MAXIMUM OF 200 SAMPLE
C        POINTS MAY BE GENERATED WITH THE PRESENT PROGRAM DIMENSIONS.
C
C        THE OUTPUT DATA MAY BE IN EITHER CARTESIAN OR SPHERICAL
C        COORDINATES AS SELECTED BY THE USER.
C
C
      IMPLICIT REAL*8(A-H,Q-Z)
      REAL *8PHI,P
      DIMENSION S(9,200), TEMPL(200), PX(200), PXVEL(200), PXACC(200), P
     1Y(200), PYVEL(200), PYACC(200), PK(200), Y(4), F(4)
C
C
C        THE FIRST DATA CARD PROVIDED MUST CONTAIN VALUES FOR THE
C        INITIAL CONDITIONS FOR THE RANGE AND RANGE RATE. THE FORMAT IS
C        (2F10.2)
C
C
      READ (5,1) (S(I,1),I=1,6)
    1 FORMAT (8F10.2)
C
C
C        THE SYSTEM STATES ARE DEFINED AS FOLLOWS.
C
C          S(1,K)= X POSITION
C
C          S(2,K)= X VELOCITY
C
C          S(3,K)= X ACCELERATION
C
C          S(4,K)= Y POSITION
C
C          S(5,K)= Y VELOCITY
C
C          S(6,K)= Y ACCELERATION
C
C          S(7,K)= Z POSITION
C
C          S(8,K)= Z VELOCITY
C
C          S(9,K)= Z ACCELERATION
C
C        K = 1,2,3,.........,NSAM
C
```

```
      READ (5,1) RATE,V,A,XPER,BR,E,RMAN,TRKEND
C
C        THE SECOND DATA CARD PROVIDED MUST CONTAIN VALUES FOR THE
C        FOLLOWING VARIABLES IN THE ORDER LISTED.. THE FORMAT IS 8F10.2.
C
C          RATE = SAMPLING PERIOD. (SEC.)
C
C          V = SPEED OF AIRCRAFT (YARDS/SEC.).
C
C          A = AMPLITUDE OF MANEUVER (YARDS). THE COEFFICIENT OF SIN**2
C          IN THE TRAVERSE PLANE MANEUVER.
C
C          XPER = DURATION OF THE MANEUVER AS MEASURED IN RANGE (YARDS).
C
C          BR= THE BEARING ANGLE FROM WHICH THE TARGET IS TO APPROACH
C          THE ORIGIN. (DEGREES)  ONLY POSITIVE ANGLES MAY BE USED.
C
C          E = THE ELEVATION ANGLE FROM WHICH THE TARGET IS TO APPRCACH
C          THE ORIGIN. (DEGREES)  ONLY FIRST QUADRANT ANGLES MAY BE USED
C
C          RMAN = RANGE AT WHICH MANEUVER IS TO BEGIN. (YARDS)
C
C          TRKEND= RANGE AT WHICH THE TRACK IS TO END IF THE TRACK IS TO
C          BE RANGE LIMITED. IF THE TRACK IS TO BE LIMITED BY THE NUMBER
C          OF SAMPLES, THEN TRKEND MAY BE SET TO A LARGE NEGATIVE VALUE.
C
C        THE EQUATION...MAX ACCELERATION = 2*A*(V**2)*((2*PI/XPER)**2)..
C        MAY BE USEFUL IN DETERMINING VALUES FOR V, A, AND XPER.
C
C
      READ (5,1) CPA,HO
C
C        THE THIRD DATA CARD PROVIDED MUST CONTAIN VALUES FOR THE
C        FOLLOWING VARIABLES, IN THE ORDER LISTED...(FORMAT 8F10.2.)
C
C          CPA = THE DISPLACEMENT IN THE Y-DIRECTION WHEN X=0.0 (YARDS).
C
C          HO = THE DISPLACEMENT IN THE Z-DIRECTION WHEN X=0.0 (YARDS).
C          THE DISPLACEMENT MUST BE IN THE POSITIVE Z-DIRECTION.
C
C        *** CAUTION *** THE USE OF NON-ZERO VALUES FOR CPA AND/OR HO
C        WILL RESULT IN A SHIFT IN THE TARGET POSITION WHICH MAY BE
C        UNDESIRABLE. FOR EXAMPLE, IF THE RANGES SPECIFIED FOR RMAN AND
C        TRKEND ARE CRITICAL THEN SPECIAL PRECAUTIONS MUST BE TAKEN IF
C        CPA OR HO IS NON-ZERO.
C
C
      READ (5,2) NSAM,N,ICOORD,IPUNCH
```

```
    2 FORMAT (4I10)
C
C
C         THE FOURTH DATA CARD PROVIDED MUST CONTAIN VALUES FOR THE
C         FOLLOWING FLAGS AND INTEGER VARIABLES IN THE ORDER LISTED.....
C         THE FORMAT IS 4I10.
C
C         NSAM = THE NUMBER OF SAMPLES TO BE GENERATED IF THE TRACK IS TO
C         BE LIMITED BY THE NUMBER OF SAMPLES. IF THE TRACK IS TO BE
C         RANGE LIMITED, SET NSAM TO SOME LARGE VALUE (E.G. NSAM = 200).
C
C         N = THE NUMBER OF STATES TO BE GENERATED. N = 6 FOR A TRACK IN
C         THE X-Y (RANGE-BEARING) PLANE. N = 9 FOR A 3-DIMENSIONAL TRACK.
C
C         ICOORD IS A FLAG. IF ICOORD IS NOT ZERC, THE TRACK OUTPUT WILL
C         BE IN SPHERICAL COORDINATES. THE SPHERICAL COORDINATE OUTPUT
C         HAS UNITS OF YARDS IN RANGE ANC RADIANS IN BEARING AND ELEV.
C
C         IPUNCH IS A FLAG. IF IPUNCH IS NOT ZERO THE TRAJECTORY WILL BE
C         PUNCHED CN DATA CARDS.
C         THE PUNCHED CARD OUTPUT WILL BE DETERMINED BY STATEMENTS
C         IMMECIATELY PRECEDING STATEMENT NUMBER 17 OF THIS PROGRAM.
C
      NT = 1
      K = 1
C
C           THE FOLLOWING SECTION GENERATES A STRAIGHT SECTION OF TRACK
C         BEGINNING AT S(1,1) AND ENDING AT RMAN (CR WHEN K = NSAM IF A
C         STRAIGHT TRACK IS DESIRED).
C
    3 K = K+1
      M = K-1
C
C
C
      DC 4 I=3,6
    4 S(I,K) = 0.0
C
C
C
      S(2,K) = -V
      S(1,K) = S(1,M)+S(2,M)*RATE
      IF((K.GE.NSAM).OR.(S(1,K).LE.TRKEND)) GO TO 10
      IF (S(1,K).LE.RMAN) GO TO 5
      GC TO 3
C
C           THE FCLLOWING SECTION OF THE PROGRAM GENERATES THE SIN**2
C           MANEUVER, BEGINNING AT RMAN AND HAVING A DURATICN = XPER.
```

```
C
    5 Y(1) = XPER
      Y(2) = S(2,K)
      Y(3) = S(4,K)
      Y(4) = S(5,K)
      X = 0.0
      H = 0.01
      SAMP = 0.0
      B = 6.283185308/XPER
      KHOLD = K
      NT = 0
    6 COS1 = DCOS(B*Y(1))
      SIN1 = DSIN(B*Y(1))
      COSSQ = COS1**2
      SINSQ = SIN1**2
      NSIGN = 2.0*Y(1)/XPER
      P = (-1.0)**NSIGN
      F(1) = Y(2)
      F(2) = -4.*(A**2)*(B**3)*Y(2)*V*SIN1*COS1*(1.-2.*COSSQ)/(1.+4.*(A*
     1*2)*(B**2)*COSSQ*SINSQ)**1.50
      F(3) = Y(4)
      F(4) = 2.0*P*A*B*(F(2)*COS1*SIN1-B*(Y(2)**2)*(1.0-2.*COSSQ))
      IF ((RATE-SAMP).GT.0.0001) GO TO 7
      K = K+1
      S(1,K) = S(1,KHOLD)+Y(1)-XPER
      S(2,K) = Y(2)
      S(3,K) = F(2)
      S(4,K) = Y(3)
      S(5,K) = Y(4)
      S(6,K) = F(4)
      IF (K.GE.NSAM) GO TO 10
      IF (Y(1).LE.0.0) GO TO 8
      SAMP = 0.0
    7 CONTINUE
      G = RKLDEQ(4,Y,F,X,H,NT)
      IF (G.LT.1.5) GO TO 6
      SAMP = SAMP+H
      GO TO 6
C
C
C       THE FOLLOWING SECTION GENERATES A STRAIGHT SECTION OF TRACK
C       WHICH EXTENDS FROM THE COMPLETION OF THE MANEUVER TO TRKEND.
C
C
    8 K = K+1
      M = K-1
C
C
```

```
C
      DC 9 I=3,6
    9 S(I,K) = 0.0
C
C
C
      S(2,K) = -V
      S(1,K) = S(1,M)+S(2,M)*RATE
      IF (K.GE.NSAM) GO TO 10
      IF (S(1,K).LE.TRKEND) GO TO 10
      GC TO 8
C
C         THE FCLLOWING SECTION ROTATES THE TRACK THROUGH THE ANGLES BR
C        AND E AS REQUIRED BY THE INPUT DATA.
C
   10 B = BR/57.29577951
      E = E/57.29577951
      CCSB = DCOS(B)
      SINB = CSIN(B)
      CCSE = DCOS(E)
      SINE = DSIN(E)
C
C
C
      DC 13 KL=1,K
C
C
C
      DC 11 JL=1,6
   11 TEMPL(JL) = S(JL,KL)
C
C
      IF (N.NE.9) GO TO 12
      S(7,KL) = TEMPL(1)*SINE +HO
      S(8,KL) = TEMPL(2)*SINE
      S(9,KL) = TEMPL(3)*SINE
   12 IF (N.NE.9) COSE = 1.0
C
      R = DSQRT(TEMPL(1)**2+TEMPL(4)**2)
      THETA = DATAN2(TEMPL(4),TEMPL(1))
      PHI = THETA+B
      S(1,KL) = R*DCOS(PHI)*COSE
      S(2,KL) = (TEMPL(2)*COSB-TEMPL(5)*SINB)*COSE
      S(3,KL) = (TEMPL(3)*COSB-TEMPL(6)*SINB)*COSE
      S(4,KL) = R*COSE*DSIN(PHI)+CPA
      S(5,KL) = TEMPL(5)*COSB+TEMPL(2)*SINB*COSE
      S(6,KL) = TEMPL(6)*CCSB+TEMPL(3)*SINB*COSE
   13 CCNTINUE
```

```
C
C
C
C        CALL SUBROUTINE ICOORD IF SPHERICAL COORDINATE DATA IS DESIRED.
C
      IF (ICOORD.EQ.0) GO TO 14
      CALL COORD (S,N,K)
C
   14 IF (ICOCRD.EQ.0) WRITE (6,15)
      IF (ICOORD.NE.0) WRITE (6,16)
   15 FCRMAT ('1',' X-POSITION  X-VELOCITY  X-ACCELER   Y-POSITION  Y-VE
     1LOCITY  Y-ACCELER   Z-POSITION  Z-VELOCITY  Z-ACCELER ',/)
   16 FCRMAT ('1',5X,'RANGE',5X,'RDOT',8X,'R2DOT',8X,'BEARING',5X,'BDOT'
     1,8X,'B2DOT',5X,'ELEVATION',5X,'EDOT',8X,'E2DOT',/)
C
C
C
      DC 19 J=1,K
C
C        ************************************************************
      IF (IPUNCH.EQ.0) GO TO 18
C
C        WRITE STATEMENTS FOR PUNCHED CARD DATA OUTPUT MAY BE ADDED
C        HERE.
      WRITE (7,17) S(1,J),S(2,J),S(3,J),S(4,J)
      WRITE (7,17) S(5,J),S(6,J),S(7,J),S(8,J)
      WRITE (7,17) S(9,J)
C
   17 FCRMAT (4(1PD20.12))
   18 CCNTINUE
C
C        ************************************************************
C
      PK(J) = J
      PX(J) = S(1,J)
      PY(J) = S(4,J)
C
C
   19 WRITE (6,20) (S(I,J),I=1,N),J
C
C
C
   20 FCRMAT (' ',9G12.4,5X,I5)
      IF (NT.EQ.1) GO TO 23
      WRITE (6,21)
   21 FCRMAT ('1')
      CALL PLOTP (PX,PY,K,0)
      WRITE (6,22)
```

```
   22 FCRMAT ('0',T50,' Y(X,T)  VS X(T) ')
   23 STCP
      END
      SLBROUTINE COORD (S,N,NSAM)
C
C        THIS SUBROUTINE CONVERTS CARTESIAN COORDINATE TRAJECTCRY
C        POINTS TO SPHERICAL COORDINATES.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(N,NSAM)
C
C
C
      DC 2 K=1,NSAM
      X = S(1,K)
      Y = S(4,K)
      Z = 0.0
      VX = S(2,K)
      VY = S(5,K)
      VZ = 0.0
      AX = S(3,K)
      AY = S(6,K)
      AZ = 0.0
      IF (N.LT.9) GO TO 1
      Z = S(7,K)
      VZ = S(8,K)
      AZ = S(9,K)
    1 CCNTINUE
      R = DSQRT(X**2+Y**2+Z**2)
      B = DATAN2(Y,X)
      E = DARSIN(Z/R)
      RDCT = (VX*X+VY*Y+VZ*Z)/R
      XYSQ = X**2+Y**2
      IF (XYSQ.LT.0.0001) XYSQ=0.0001
      BDCT = (X*VY-VX*Y)/XYSQ
      RTRZ = DSQRT(R**2-Z**2)
      IF (RTRZ.LT.0.0001) RTRZ=0.0001
      EDCT = (R*VZ-RDOT*Z)/(R*RTRZ)
      R2DOT = (R*(AX*X+VX**2+AY*Y+VY**2+AZ*Z+VZ**2)-RDOT*(VX*X+VY*Y+VZ*Z
     1))/R**2
      B2DOT = (X*AY-AX*Y)/XYSQ-2*((X*VX+Y*VY)*(X*VY-VX*Y))/XYSQ**2
      E2DOT = (R*AZ-R2DOT*Z)/RTRZ-(RDOT*RTRZ**2+R*(RDOT*R-VZ*Z)*(R*VZ-RD
     1OT*Z))/(R*RTRZ**3)
      S(1,K) = R
      S(2,K) = RDOT
      S(3,K) = R2DOT
      S(4,K) = B
      S(5,K) = BDCT
```

```
      S(6,K) = B2DOT
      IF (N.LT.9) GO TO 2
      S(7,K) = E
      S(8,K) = EDOT
      S(9,K) = E2DOT
    2 CCNTINUE
C
C
C
      WRITE (6,3)
    3 FCRMAT ('1',T30,' OUTPUT DATA IN SPHERICAL COORDINATES',/)
      RETURN
      END
```

APPENDIX D

ADAPTIVE FILTER SUBROUTINES

FORTRAN listings of the three adaptive filter subroutines are included in this appendix. Also included is a flowchart which is based on the first version of the switch-on-residuals adaptive filter subroutine, because there is essentially no difference in the program flow, there is no separate flowchart provided for the second version.

Switch-On-Range Adaptive Filter

```
      SUBROUTINE ADAPT
C
C        THIS IS THE SWITCH-ON-RANGE VERSION OF SUBROUTINE ADAPT.
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
C
C     THE USER MUST SUPLY STATEMENTS HERE TO CONTROL THE ADAPTIVE
C     FILTER OPERATION.
C
C
C        CALCULATE THE TARGET RANGE FROM THE CARTESIAN MEASUREMENTS.
C
      RNG = SQRT(Z(1)**2+Z(2)**2+Z(3)**2)
C
C        IF THE TARGET RANGE IS LESS THAN THE SWITCHING RANGE SWITCH TO
C        THE HIGH Q GAIN MATRIX.
C
      IF ((RNG.GT.3333.33).OR.(K.LT.3)) GO TO 2
C
    1 GK(1,1) = 0.250569
      GK(2,1) = 0.144287
      GK(4,2) = 0.719631
      GK(5,2) = 1.42079
      GK(7,3) = 0.719631
      GK(8,3) = 1.42079
    2 RETURN
      END
```

Switch-On-Residuals Adaptive Filter No. 1

```
      SUBROUTINE ADAPT
C        THIS SUBROUTINE CONTROLS THE ADAPTIVE SWITCHING OF THE FILTER
C        PARAMETERS IN ORDER TO MAKE THE FILTER MORE RESPONSIVE IN A
C        CHANGING SITUATION. THIS IS THE SWITCH-ON-RANGE ADAPTIVE
C        FILTER SUBROUTINE.
C
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
      DIMENSION GKA(9,9,41)
C
C     THE USER MUST SUPLY STATEMENTS HERE TO CONTROL THE ADAPTIVE
C     FILTER OPERATION.
C
      IF (K-3) 1,2,6
C
C        THE FILTER IS NOT ADAPTIVE FOR THE FIRST THREE SAMPLES.
C
    1 RETURN
    2 IF (ITER.GT.1) GO TO 5
C
C        ON THE FIRST ITERATION READ IN THE MANEUVERING FILTER GAIN
C        SCHEDULE.
C
C
      DO 3 KK=1,40
C
      DO 3 I=1,N
    3 READ (5,4) (GKA(I,J,KK),J=1,M)
C
    4 FORMAT (4F20.12)
    5 CONTINUE
      IFLAGX = 0
      IFLAGY = 0
      IFLAGZ = 0
      IBX = 0
      IBY = 0
      IBZ = 0
C
C        SET IN THE STEADY STATE THEORETICAL VARIANCE OF POSITION
C        ESTIMATE ERROR VALUES TO BE USED IN CALCULATING THE GATING
C        LEVELS.
```

```
C
      PX = 2.655313
      PY = 2.93339
      PZ = 2.93339
C
C        IF THE GATING FLAGS ARE ZERO RESET THE MANEUVERING GAIN
C        SCHEDULE INDICES TO THEIR INITIAL VALUES.
C
    6 IF (IFLAGX.EQ.0) KX=0
      IF (IFLAGY.EQ.0) KY=0
      IF (IFLAGZ.EQ.0) KZ=0
      RSQ = Z(1)**2+Z(2)**2+Z(3)**2
      SINSQE = Z(3)**2/RSQ
      COSSQE = 1.0-SINSQE
      SIGSQR = SIGV(1)**2
      SIGSQB = SIGV(2)**2
      SIGSQE = SIGV(3)**2
C
C
C        CALCULATE THE VARIANCE OF MEASUREMENT NOISE USING EQ.(2.14)
C
      RKSQX = SIGSQR+RSQ*SIGSQE*SINSQE
      RKSQY = SIGSQB*RSQ*(COSSQE+SIGSQE*SINSQE)
      RKSQZ = SIGSQE*RSQ*(COSSQE+SIGSQR*SINSQE)
C
C        CALCULATE THE GATING LEVELS.
C
      BGX = PX+RKSQX
      GATEX = 4.0*BGX
      BGY = PY+RKSQY
      GATEY = 4.0*BGY
      BGZ = PZ+RKSQZ
      GATEZ = 4.0*BGZ
C
C        CALCULATE THE SAMPLE RESIDUALS.
C
      RESX = (Z(1)-XHKKM1(1))**2
      RESY=(Z(2)-XHKKM1(4    ))**2
      RESZ=(Z(3)-XHKKM1(   7    ))**2
      IF (RESX.LT.GATEX) GO TO 7
C
C        IF THE X-POSITION RESIDUAL IS GREATER THAN THE X-POSITION
C        GATING LEVEL SET GATING FLAG IFLAGX.
C
      IFLAGX = IFLAGX+1
      IBX = 0
      GO TO 8
C
```

```
C        IF THE X-POSITION RESIDUAL IS LESS THAN ONE HALF THE GATING
C        LEVEL SET RESET FLAG IBX.
C
    7 IF (RESX.LT.BGX) IBX=IBX+1
C
    8 IF (RESY.LT.GATEY) GO TO 9
C
C        IF THE Y-POSITION RESIDUAL IS GREATER THAN THE Y-POSITION
C        GATING LEVEL SET GATING FLAG IFLAGY.
C
      IFLAGY = IFLAGY+1
      IBY = 0
      GC TO 10
C
C        IF THE Y-POSITION RESIDUAL IS LESS THAN ONE HALF THE GATING
C        LEVEL SET RESET FLAG IBY.
C
    9 IF (RESY.LT.BGY) IBY=IBY+1
   10 IF (RESZ.LT.GATEZ) GO TO 11
C
C        IF THE Z-POSITION RESIDUAL IS GREATER THAN THE Z-POSITION
C        GATING LEVEL SET GATING FLAG IFLAGZ.
C
      IFLAGZ = IFLAGZ+1
      IBZ = 0
      GC TO 12
C
C        IF THE Z-POSITION RESIDUAL IS LESS THAN ONE HALF THE GATING
C        LEVEL SET RESET FLAG IBZ.
C
   11 IF (RESZ.LT.BGZ) IBZ=IBZ+1
C
C        IF A RESIDUAL HAS BEEN LESS THAN ONE HALF THE GATING LEVEL
C        FOR THREE SUCCESSIVE SAMPLES RESET THE CORRESPONDING GATING
C        FLAG TO ZERO.
C
   12 IF (IBX.GE.3) IFLAGX=0
      IF (IBY.GE.3) IFLAGY=0
      IF (IBZ.GE.3) IFLAGZ=0
C
C        IF THE X-DIRECTION GATING FLAG, IFLAGX IS GREATER THAN ZERO
C        FOR TWO SUCCESSIVE SAMPLES SWITCH TO THE MANEUVERING GAIN
C        SCHEDULE FOR THE X-DIRECTION ESTIMATES.
C
      IF (IFLAGX.LE.1) GO TO 14
      KX = KX+1
      IF (KX.GT.39) KX = 40
C
```

```
      DC 13 I=1,N3
   13 GK(I,1) = GKA(I,1,KX)
C
C
C
C        IF THE Y-DIRECTION GATING FLAG, IFLAGY IS GREATER THAN ZERO
C        FOR TWO SUCCESSIVE SAMPLES SWITCH TO THE MANEUVERING GAIN
C        SCHEDULE FOR THE Y-DIRECTION ESTIMATES.
C
   14 IF (IFLAGY.LE.1) GO TO 16
      KY = KY+1
      IF (KY.GT.39) KY = 40
C
      DC 15 I=4,6
   15 GK(I,2) = GKA(I,2,KY)
C
C
C
C        IF THE Z-DIRECTION GATING FLAG, IFLAGZ IS GREATER THAN ZERO
C        FOR TWO SUCCESSIVE SAMPLES SWITCH TO THE MANEUVERING GAIN
C        SCHEDULE FOR THE Z-DIRECTION ESTIMATES.
C
   16 IF (IFLAGZ.LE.1) GO TO 18
      KZ = KZ+1
      IF (KZ.GT.39) KZ = 40
C
      DO 17 I=7,9
   17 GK(I,3) = GKA(I,3,KZ)
C
   18 RETURN
      END
```

Switch-On-Residuals Adaptive Filter No. 2

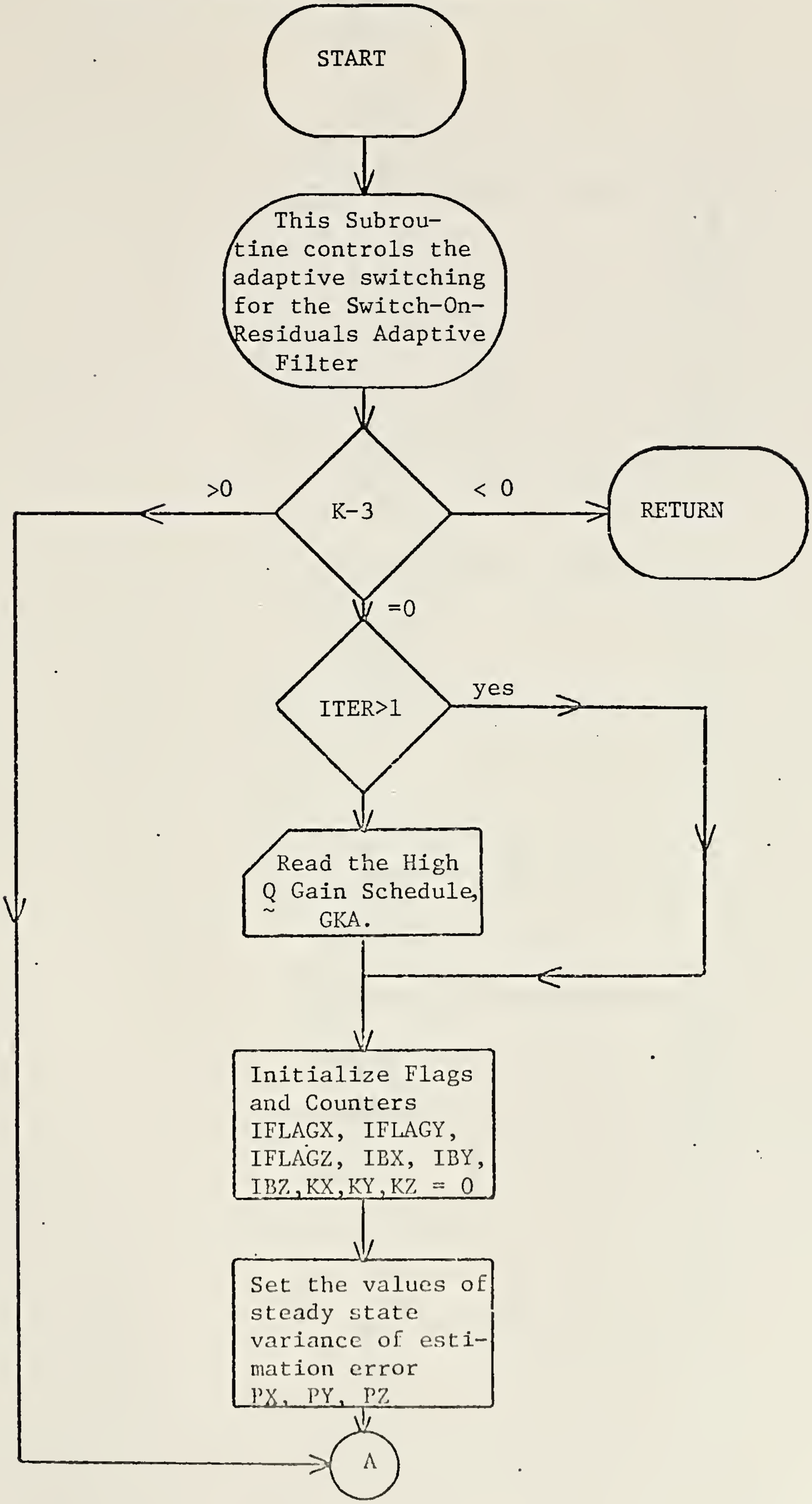
```
      SUBROUTINE ADAPT
C        THIS SUBROUTINE CONTROLS THE ADAPTIVE SWITCHING OF THE FILTER
C        PARAMETERS IN ORDER TO MAKE THE FILTER MORE RESPONSIVE IN A
C        CHANGING SITUATION. THIS IS THE SWITCH-ON-RESIDUALS ADAPTIVE
C        FILTER SUBROUTINE.
C
C
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(9,9),Q(9,9),G(9,9),PKK(9,9),GAMMA(9,9),COVW(9,9),TEMP(9,
     19),TEMP1(9,9),TEMP2(9,9),H(9,9),PKKM1(9,9),R(9,9),PHI(9,9),VAR(9,9
     2,160),GKS(9,9,160),PKKS(9,9,160),XM(9,160),ERR(9,160),GAMMAS(9,9),
     3PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGW(9),X(9),SIGXZ(9),XZMEAN(9
     4),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SIGV(9),XHATZ(9),AMISS(160),
     5N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NFIRE,NHIT
     6,NSCARE,ICOORD,IADAPT
      DIMENSION GKA(9,9,41)
C
C     THE USER MUST SUPLY STATEMENTS HERE TO CONTROL THE ADAPTIVE
C     FILTER OPERATION.
C
      IF (K-13) 1,2,6
C
C        THE FILTER IS NOT ADAPTIVE FOR THE FIRST THIRTEEN SAMPLES.
C
    1 RETURN
    2 IF (ITER.GT.1) GO TO 5
C
C        ON THE FIRST ITERATION READ IN THE MANEUVERING FILTER GAIN
C        SCHEDULE.
C
C
      DO 3 KK=1,40
C
      DO 3 I=1,N
    3 READ (5,4) (GKA(I,J,KK),J=1,M)
C
    4 FORMAT (4F20.12)
    5 CONTINUE
      IFLAGX = 0
      IFLAGY = 0
      IFLAGZ = 0
      IBX = 0
      IBY = 0
      IBZ = 0
      KX = 0
      KY = 0
      KZ = 0
C
```

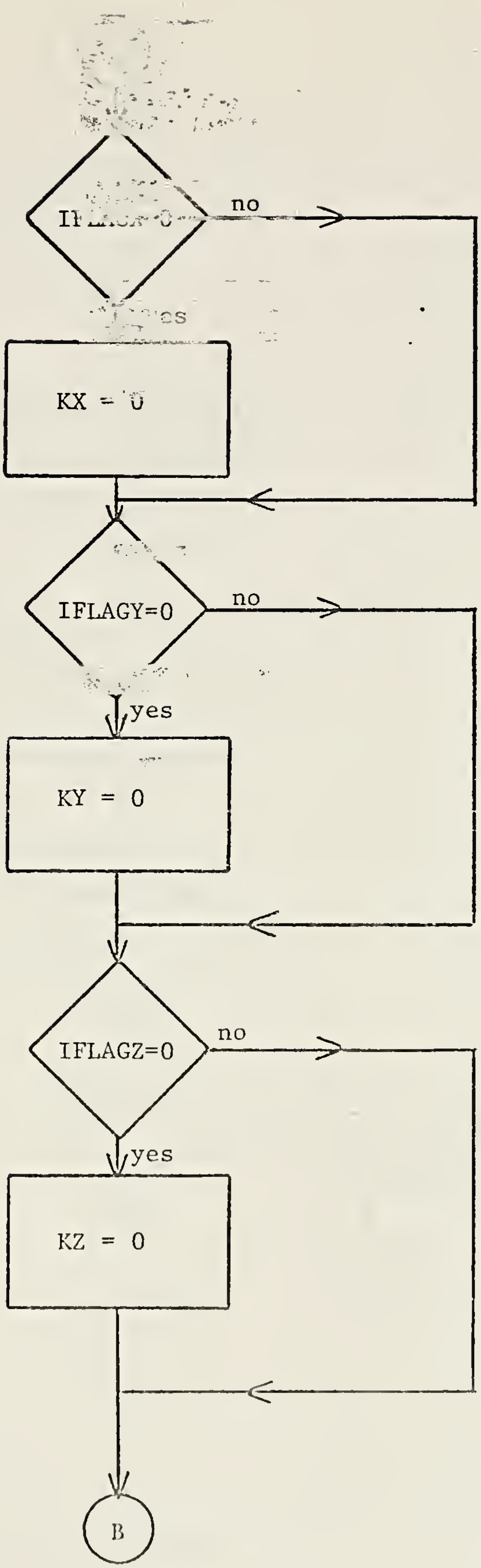
```
C
C        SET IN THE STEADY STATE THEORETICAL VARIANCE OF POSITION
C        ESTIMATE ERROR VALUES TO BE USED IN CALCULATING THE GATING
C        LEVELS.
C
      PX = 2.655313
      PY = 2.93339
      PZ = 2.93339
C
C        IF THE GATING FLAGS ARE ZERO RESET THE MANEUVERING GAIN
C        SCHEDULE INDICES TO THEIR INITIAL VALUES.
C
    6 IF ((IFLAGX.EQ.0).AND.(IBX.GT.3)) KX=0
      IF ((IFLAGY.EQ.0).AND.(IBY.GT.3)) KY=0
      IF ((IFLAGZ.EQ.0).AND.(IBZ.GT.3)) KZ=0
      RSQ = Z(1)**2+Z(2)**2+Z(3)**2
      SINSQE = Z(3)**2/RSQ
      COSSQE = 1.0-SINSQE
      SIGSQR = SIGV(1)**2
      SIGSQB = SIGV(2)**2
      SIGSQE = SIGV(3)**2
C
C
C        CALCULATE THE VARIANCE OF MEASUREMENT NOISE USING EQ.(2.14)
C
      RKSQX = SIGSQR+RSQ*SIGSQE*SINSQE
      RKSQY = SIGSQB*RSQ*(COSSQE+SIGSQE*SINSQE)
      RKSQZ = SIGSQE*RSQ*(COSSQE+SIGSQR*SINSQE)
C
C        CALCULATE THE GATING LEVELS.
C
      BGX = PX+RKSQX
      GATEX = 9.0*BGX
      BGY = PY+RKSQY
      GATEY = 9.0*BGY
      BGZ = PZ+RKSQZ
      GATEZ = 9.0*BGZ
C
      BGX = 7.0*BGX
      BGY = 7.0*BGY
      BGZ = 7.0*BGZ
C        CALCULATE THE SAMPLE RESIDUALS.
C
      RESX = (Z(1)-XHKKM1(1))**2
      RESY = (Z(2)-XHKKM1(4))**2
      RESZ = (Z(3)-XHKKM1(7))**2
      IF (RESX.LT.GATEX) GO TO 7
C
C        IF THE X-POSITION RESIDUAL IS GREATER THAN THE X-POSITION
```

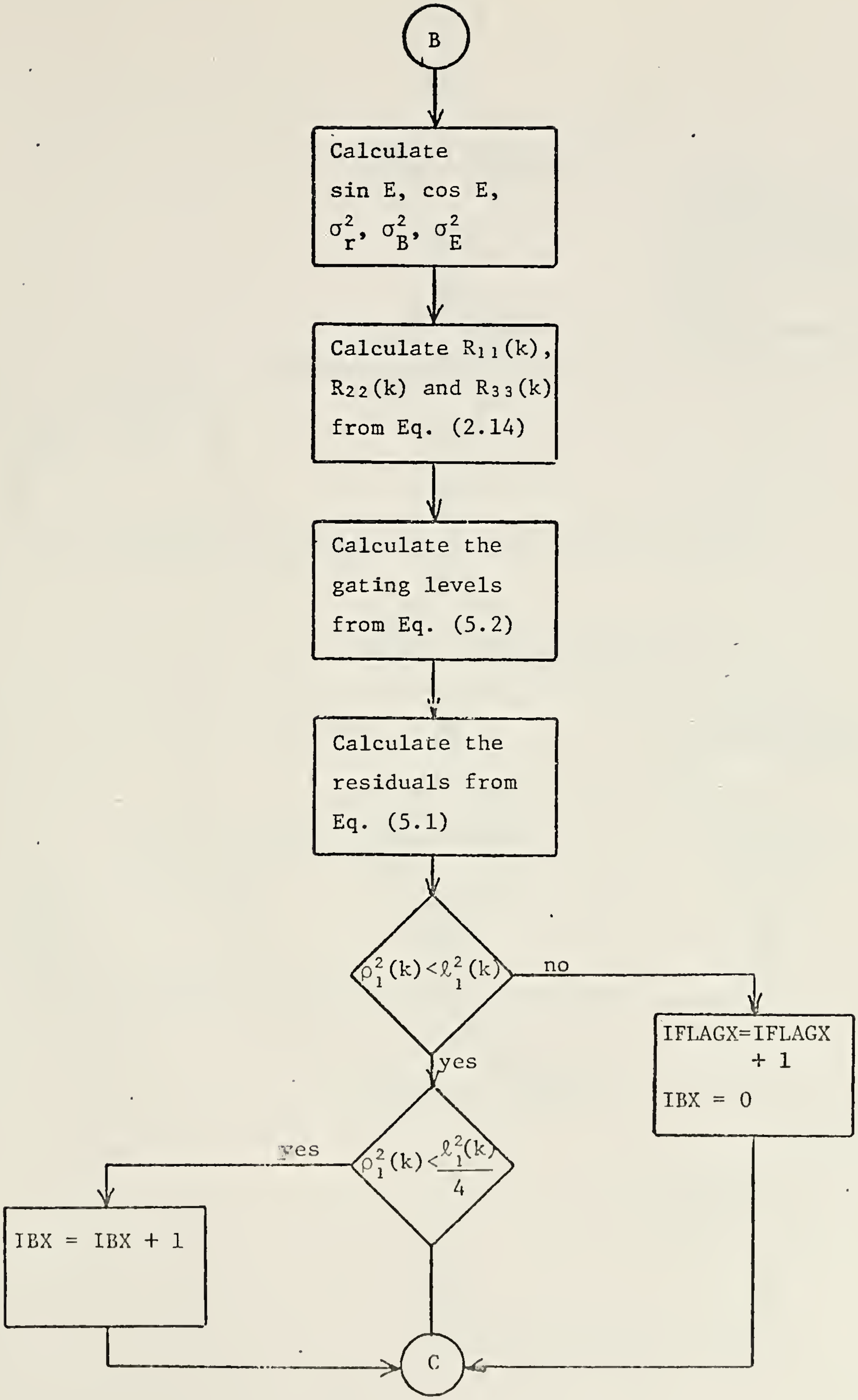
```
C           GATING LEVEL SET GATING FLAG IFLAGX.
C
         IFLAGX = IFLAGX+1
         IBX = 0
         GO TO 8
C
C           IF THE X-POSITION RESIDUAL IS LESS THAN 88.0/100 THE GATING
C           LEVEL SET RESET FLAG IBX.
C
       7 IF (RESX.LT.BGX) IBX=IBX+1
C
       8 IF (RESY.LT.GATEY) GO TO 9
C
C           IF THE Y-POSITION RESIDUAL IS GREATER THAN THE Y-POSITION
C           GATING LEVEL SET GATING FLAG IFLAGY.
C
         IFLAGY = IFLAGY+1
         IBY = 0
         GO TO 10
C
C           IF THE Y-POSITION RESIDUAL IS LESS THAN 88.0/100 THE GATING
C           LEVEL SET RESET FLAG IBY.
C
       9 IF (RESY.LT.BGY) IBY=IBY+1
      10 IF (RESZ.LT.GATEZ) GO TO 11
C
C           IF THE Z-POSITION RESIDUAL IS GREATER THAN THE Z-POSITION
C           GATING LEVEL SET GATING FLAG IFLAGZ.
C
         IFLAGZ = IFLAGZ+1
         IBZ = 0
         GO TO 12
C
C           IF THE Z-POSITION RESIDUAL IS LESS THAN  88/100  THE GATING
C           LEVEL SET RESET FLAG IBZ.
C
      11 IF (RESZ.LT.BGZ) IBZ=IBZ+1
C
C           IF A RESIDUAL HAS BEEN LESS THAN ONE HALF THE GATING LEVEL
C           FOR TWO   SUCCESSIVE SAMPLES RESET THE CORRESPONDING GATING
C           FLAG TO ZERO.
C
      12 IF (IBX.GE.2) IFLAGX=0
         IF (IBY.GE.2) IFLAGY=0
         IF (IBZ.GE.2) IFLAGZ=0
C
C           IF THE X-DIRECTION GATING FLAG, IFLAGX IS GREATER THAN ZERO
C           SWITCH TO THE MANEUVERING GAIN SCHEDULE FOR  THE X-DIRECTION
```

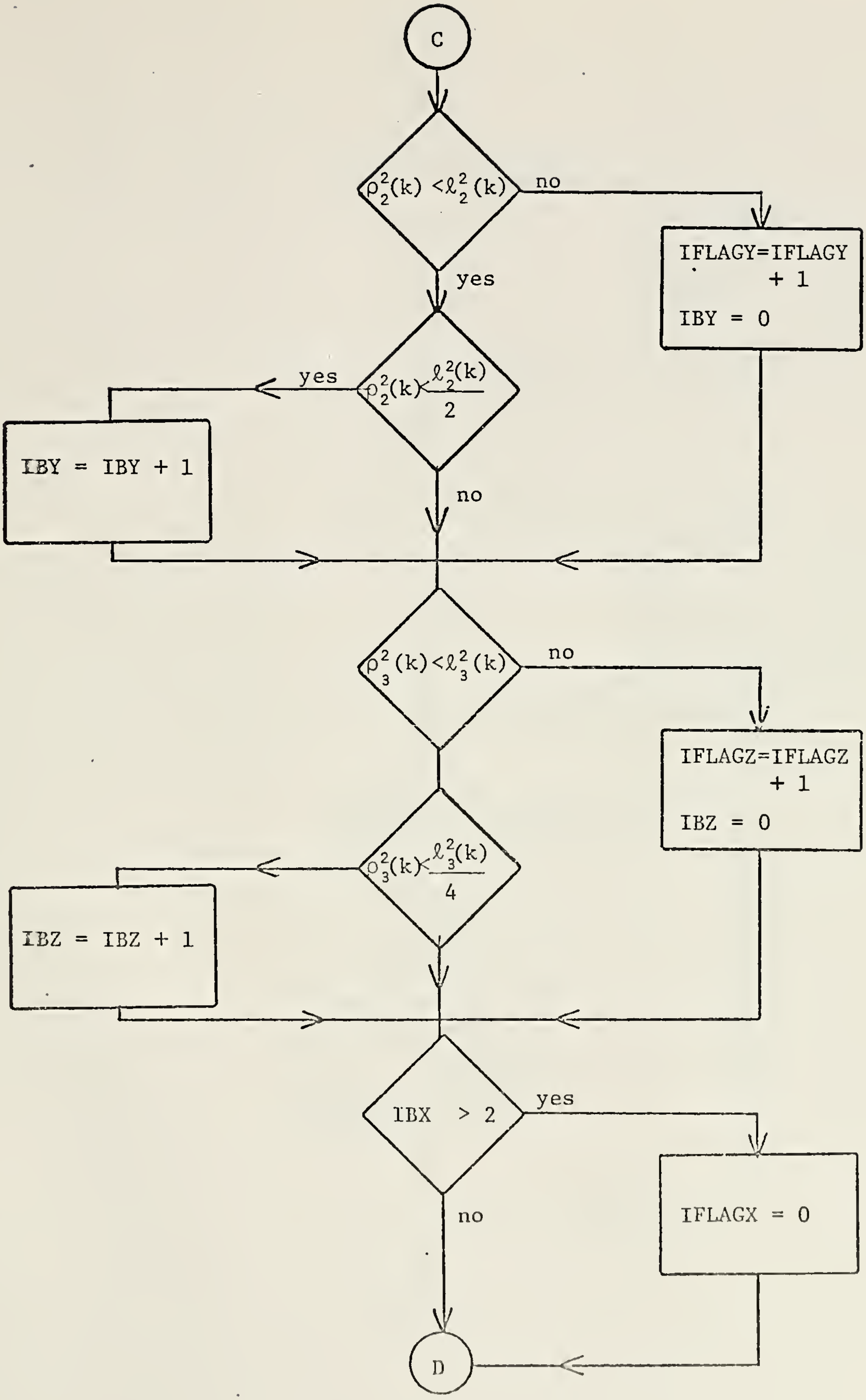
```
C          STATE ESTIMATES.
C
      IF (IFLAGX.LT.1) GO TO 14
      KX = KX+1
      IF (KX.GT.39) KX = 40
C
      DC 13 I=1,N3
   13 GK(I,1) = GKA(I,1,KX)
C
C
C
C          IF THE Y-DIRECTION GATING FLAG, IFLAGY IS GREATER THAN ZERO
C          SWITCH TO THE MANEUVERING GAIN SCHEDULE FOR  THE Y-DIRECTICN
C          STATE ESTIMATES.
C
   14 IF (IFLAGY.LT.1) GO TO 16
      KY = KY+1
      IF (KY.GT.39) KY = 40
C
      DC 15 I=4,6
   15 GK(I,2) = GKA(I,2,KY)
C
C
C
C          IF THE Z-DIRECTION GATING FLAG, IFLAGZ IS GREATER THAN ZERO
C          SWITCH TO THE MANEUVERING GAIN SCHEDULE FOR  THE Z-DIRECTICN
C          STATE ESTIMATES.
C
   16 IF (IFLAGZ.LT.1) GC TO 18
      KZ = KZ+1
      IF (KZ.GT.39) KZ = 40
C
      DC 17 I=7,9
   17 GK(I,3) = GKA(I,3,KZ)
C
   18 RETURN
      END
```

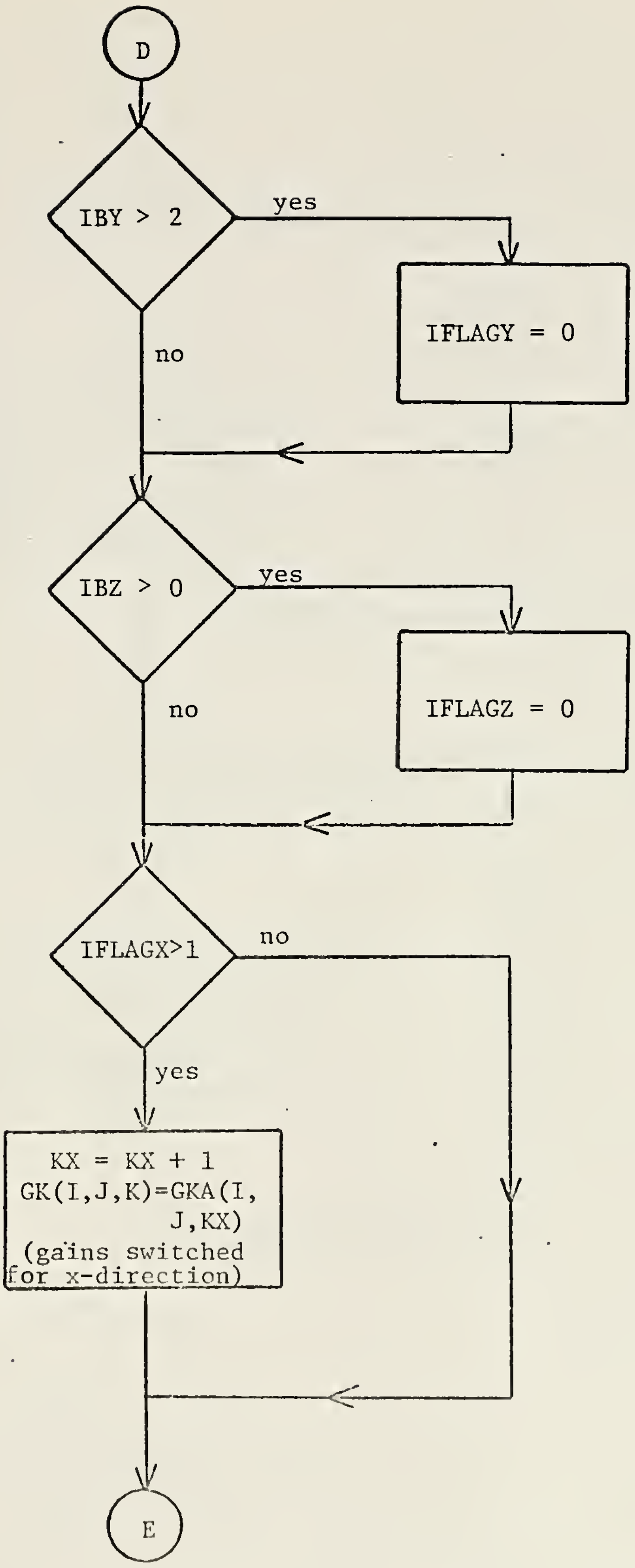
SWITCH-ON-RESIDUALS SUBROUTINE FLOWCHART

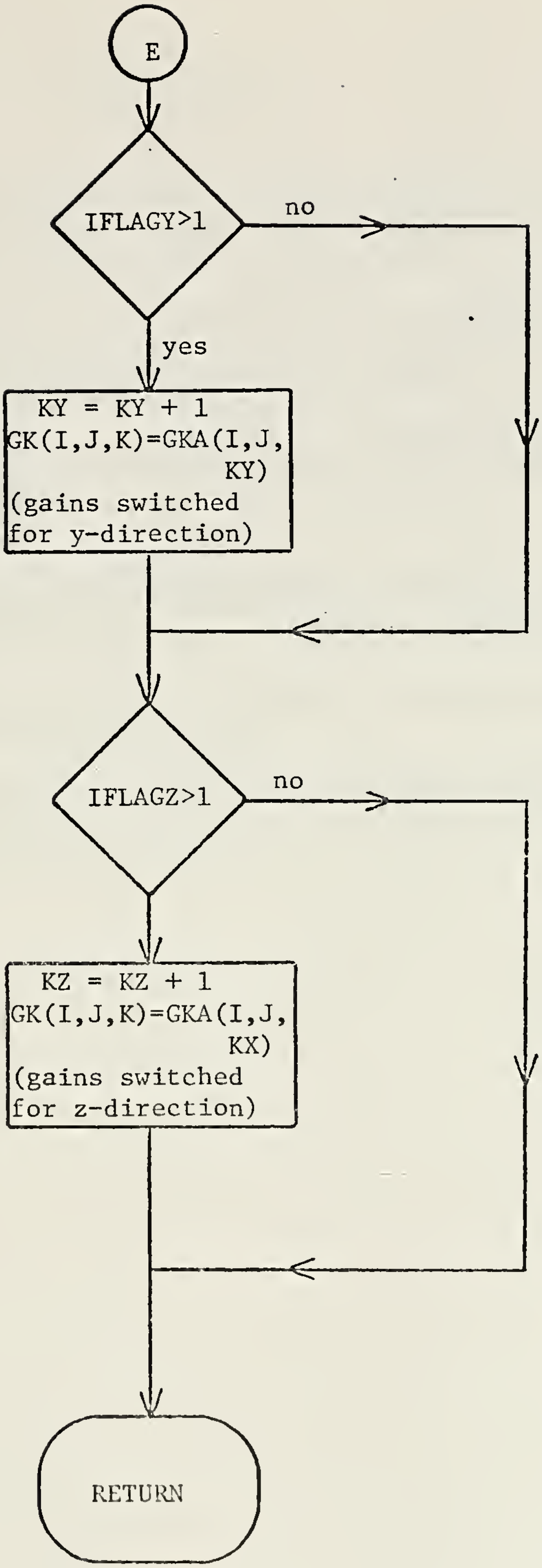

START
This Subroutine controls the adaptive switching for the Switch-On-Residuals Adaptive Filter
K-3
>0
< 0
RETURN
=0
ITER>1
yes
Read the High Q Gain Schedule, GKA.
Initialize Flags and Counters IFLAGX, IFLAGY, IFLAGZ, IBX, IBY, IBZ, KX, KY, KZ = 0
Set the values of steady state variance of estimation error PX, PY, PZ
A


IFLAGX=0
no
yes
KX = 0
IFLAGY=0
no
yes
KY = 0
IFLAGZ=0
no
yes
KZ = 0
B


B
Calculate sin E, cos E, σ_r^2, σ_B^2, σ_E^2
Calculate $R_{11}(k)$, $R_{22}(k)$ and $R_{33}(k)$ from Eq. (2.14)
Calculate the gating levels from Eq. (5.2)
Calculate the residuals from Eq. (5.1)
$\rho_1^2(k) < \ell_1^2(k)$
no
IFLAGX=IFLAGX + 1
IBX = 0
yes
$\rho_1^2(k) < \frac{\ell_1^2(k)}{4}$
yes
IBX = IBX + 1
C


C
$\rho_2^2(k) < \ell_2^2(k)$
no
IFLAGY=IFLAGY + 1
IBY = 0
yes
$\rho_2^2(k) < \frac{\ell_2^2(k)}{2}$
yes
IBY = IBY + 1
no
$\rho_3^2(k) < \ell_3^2(k)$
no
IFLAGZ=IFLAGZ + 1
IBZ = 0
$\rho_3^2(k) < \frac{\ell_3^2(k)}{4}$
IBZ = IBZ + 1
IBX > 2
yes
IFLAGX = 0
no
D


D
IBY > 2
yes
no
IFLAGY = 0
IBZ > 0
yes
no
IFLAGZ = 0
IFLAGX>1
no
yes
KX = KX + 1
GK(I,J,K)=GKA(I,
J,KX)
(gains switched
for x-direction)
E


E
IFLAGY>1
no
yes
KY = KY + 1
GK(I,J,K)=GKA(I,J,KY)
(gains switched for y-direction)
IFLAGZ>1
no
KZ = KZ + 1
GK(I,J,K)=GKA(I,J,KX)
(gains switched for z-direction)
RETURN

BIBLIOGRAPHY

1. Naval Postgraduate School Report NPS-52KI74101, Evaluation of State Estimators and Predictors for Fire Control Systems, by D.E. Kirk, October 1974.

2. Ketron, M.G., Monte-Carlo Evaluation of Digital Filters for Fire Control Systems, MSEE Thesis, Naval Postgraduate School, Monterey, California, 1974.

3. Demetry, J.S., Notes on the Theory and Applications of Optimal Estimation, Naval Postgraduate School, Monterey, California, 1970.

4. McKinley, D.H., Modeling Maneuvering Airborne Target Motion With a Coloring Noise Filter, MSEE Thesis, Naval Postgraduate School, Monterey, California, 1974.

5. Jazwinski, A.H., Stochastic Processes and Filtering Theory, Academic Press, 1970.

6. Department of the Navy, Bureau of Ordnance, Ordnance Pamphlet 1184 (First Revision), 5 April 1957.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor D. E. Kirk, Code 52Ki Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	3
4.	Professor H. A. Titus, Code 52Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	LT John Michael Parr, USN Class 1-75 Naval Destroyer School Newport, Rhode Island 02840	2
6.	Mr. H. A. Parish System Integration and Engineering Division Code 4320 Port Hueneme, California 93043	1
7.	Mr. John M. Corcoran Systems Test/Evaluation Branch Code 4045 Port Hueneme, California 93043	1
8.	LT D. F. Regener Code 721 COMOPTEVFOR Norfolk, Virginia 23511	1
9.	Department Chairman, Code 52 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1

Thesis 157300
P19 Parr
c.1 An estimator for an
anti-aircraft gun fire
control system.

13 OCT 84 33148

Thesis 157300
P19 Parr
c.1 An estimator for an
anti-aircraft gun fire
control system.

thesP19
An estimator for an anti-aircraft gun fi
3 2768 001 00249 6
DUDLEY KNOX LIBRARY